

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

**Systém pro evidenci a správu
audiovizuálních souborů digitálního kina
dle specifikace DCI**

**A System for Evidence of Audiovisual
Files of Digital Cinema by DCI
Specification**

Zadání bakalářské práce

Student:

Lukáš Tatarin

Studijní program:

B2649 Elektrotechnika

Studijní obor:

2612R041 Řídicí a informační systémy

Téma:

**Systém pro evidenci a správu audiovizuálních souborů digitálního kina
dle specifikace DCI
A System for Evidence of Audiovisual Files of Digital Cinema
by DCI Specification**

Jazyk vypracování:

čeština

Zásady pro vypracování:

V rámci bakalářské práce bude student řešit možnosti automatizace technických zařízení digitálního kina dle specifikace DCI (digitální projektor, kino server, zvukový procesor, odtah strojů, atd.).

Úkolem bude najít nejvhodnější řešení pro jednotné ovládání z jednoho místa pomocí aplikace pro platformu Android. Student se dále seznámí s aktuálním stavem evidence audiovizuálních souborů (filmy, reklamy, trailery, atd.) a navrhne databázi pro jejich správu a třídění. Vezme rovněž v potaz jednotlivé vlastnosti AV souborů, jako je obrazový formát, zvukový formát, jazyková lokalizace, atd. Výsledkem by měla být reálná aplikace psaná v jazyce Java obsluhující bezdrátově jednotlivé zařízení a databázi s evidencí AV souborů.

Cíl bakalářské práce:

1. Student se seznámí s možnostmi bezdrátových komunikací a databázových systémů.
2. Provedení datové analýzy parametrů filmových titulů.
3. Vytvoření návrhu a implementace databáze filmových titulů.
4. Pro vizualizaci a komunikaci s databází vytvoří mobilní aplikaci na platformě Android.
5. Vytvoření funkčního vzorku a zhodnocení výsledků.

Seznam doporučené odborné literatury:

- [1] GERBER, Adam a Clifton CRAIG. *Learn Android Studio: build Android apps quickly and effectively*. Berkeley, CA: Apress, 2015. ISBN 978-1430266013.
- [2] POKORNÝ, Jaroslav a Ivan HALAŠKA. *Databázové systémy: vybrané kapitoly a cvičení*. 2. přeprac. vyd. Praha: Karolinum, 1998. ISBN 80-7184-687-2.
- [3] RABOWSKY, Irving. *System and method for digital electronic cinema delivery*. U.S. Patent No 6,141,530, 2000.
- [4] BLOOM, Jeffrey A. Digital cinema content security and the DCI. In: *40th Annual Conference on Information Sciences and Systems*. Princeton, NJ, USA: IEEE, 2006. p. 1176-1181. Print ISBN: 1-4244-0349-9. DOI: 10.1109/CISS.2006.286643.
- [5] GONG, Chengying, Jinghong XING a Yinbao HU. Data communication of Android mobile terminal and PHP and MySQL based on JSON [J]. *Industrial Instrumentation & Automation*. 2013, Issue: 1, p. 63-65,69. ISSN 1000-0682.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Tomáš Urbanczyk**

Datum zadání: 01.09.2018

Datum odevzdání: 30.04.2019



doc. Ing. Jiří Koziorek, Ph.D.
vedoucí katedry

prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 30. dubna 2019

.....


Rád bych na tomto místě poděkoval mému vedoucímu práce panu Ing. Tomáši Urbanczykovi za metodické vedení práce a cenné připomínky.

Dále poté Městu Hradec nad Moravicí a panu BcA. Ing. Richardu Vaculovi, vedoucímu kina Orion, za umožnění praxe a otestování zařízení v reálném provozu.

Paní Mgr. Silvii Holé za korekturu této práce.

Na závěr bych rád poděkoval mému kolegovi Lukáši Herudkovi, jenž se ujal práce nad vývojem a konstrukcí distribuovaného systému řízení jednotlivých periférií, který tuto práci volně rozšiřuje.

Abstrakt

Cílem této bakalářské práce je vytvoření aplikace pro mobilní platformy (Android) zajišťující promítačům pohodlné a komfortní bezdrátové ovládání a automatizaci projekční technologie a zařízení s projekční technikou bezprostředně souvisejících. Aplikace umožňuje individuální správu digitálního obsahu a poskytuje přehled všem pracovníkům, kteří s digitálním obsahem manipulují. Bakalářská práce se zabývá analýzou stávajícího stavu evidence a ovládání. Rovněž také zohledňuje podobně pracující a zcela profesionální systémy. Součástí práce je také plně funkční aplikace, která řeší napojení na on-line databázi a komunikaci s lokálním serverem, který prostřednictvím distribuovaných periférií přímo ovládá jednotlivá zařízení.

Klíčová slova: Android Studio, Android, MySQL, databáze, JAVA, PHP, Nette, DCI, DCP, KDM, Wi-Fi, HTTP, řídicí systém, správa obsahu

Abstract

The aim of this bachelor's thesis is to create an application for mobile platforms (Android) providing projectionists with easy and comfortable wireless control and automation of projection technology and devices directly related with projection techniques. The application also allows for individual management of the digital content and provides an overview to all staffs who work with digital content. The bachelor's thesis deals with the analysis of the current state of record keeping and controlling and it also takes into account similar working practices and full professional systems. Part of the work is also a fully functional application that resolves the connection to online databases and communication with the local server which via distributed peripherals directly controls individual devices.

Keywords: Android Studio, Android, MySQL, database, JAVA, PHP, Nette, DCI, DCP, KDM, Wi-Fi, HTTP, control system, content management

Obsah

Seznam použitých zkratk a symbolů	9
Seznam obrázků	11
Seznam tabulek	12
Seznam výpisů zdrojového kódu	13
Úvod	14
1 Současný stav	16
1.1 Správa digitálního obsahu	16
1.2 Vzdálené ovládání projekce	19
1.3 Požadavky na aplikaci	25
2 Systémy pro automatizaci a správu obsahu	27
2.1 Přehled systému pro automatizaci	27
2.2 Přehled systému pro správu obsahu (TMS)	30
3 Bezdrátová komunikace a předávání dat	33
3.1 Bezdrátová komunikace s mobilním zařízením	33
3.2 Výměna dat prostřednictvím TCP/IP protokolu	33
3.3 Serializační formát JSON	34
4 Problematika značení DCP balíčků	36
4.1 Struktura DCP balíčků	36
4.2 Systematika značení DCP balíčků a jejich atributů	36
4.3 Verzování a omezování platnosti DCP souborů	39
5 Databázový systém	41
5.1 Výhody a nevýhody jednotlivých SQL systémů	41
5.2 Databázový systém MySQL	42
5.3 Návrh struktury databáze	43
5.4 Komunikace s databází MySQL	45
6 Návrh a implementace aplikačního serveru	47
6.1 Směrování a routování	47
6.2 Zpracování požadavku	48
6.3 Databázový model	49
6.4 Model pro manipulaci s DCP	50

6.5	Model pro manipulaci s KDM	53
6.6	Model pro manipulaci s uživateli	56
7	Struktura navržené aplikace	59
7.1	Spuštění aplikace a přihlášení	59
7.2	Hlavní aktivita	61
7.3	Prohlížení a editace DCP a KDM	61
7.4	Ovládání zařízení	67
7.5	Komunikace aplikace s databázovým a řídicím serverem	68
8	Testování aplikace	74
8.1	Rychlost odezvy v komunikaci s databázovým serverem	74
8.2	Rychlost odezvy v komunikaci s řídicím serverem	75
	Závěr	76
	Literatura	77
	Přílohy	80
A	Blokové schéma instalace systému	81
B	Aktivity diagram mobilní aplikace	82
C	Video-ukázky	83
C.1	Spuštění a přihlášení uživatele do aplikace	83
C.2	Prohlížení záznamů DCP z databáze	83
C.3	Prohlížení záznamů KDM z databáze	83
C.4	Vložení nového DCP záznamu do databáze	83
C.5	Vložení nového KDM záznamu do databáze	83
C.6	Úprava stávajícího DCP záznamu v databázi	83
C.7	Úprava stávajícího KDM záznamu v databázi	83
C.8	Odstranění stávajícího DCP záznamu z databáze	84
C.9	Odstranění stávajícího KDM záznamu z databáze	84
C.10	Ovládání zařízení	84
D	Zdrojové kódy a instalační soubory	85
D.1	Skript pro vytvoření databáze	85
D.2	Zdrojový kód aplikačního serveru	85
D.3	Zdrojový kód mobilní aplikace	85
D.4	Instalační soubor mobilní aplikace	85

Seznam použitých zkratek a symbolů

ACID	– Atomic, Consistent, Isolated, Durable
AES	– Audio Engineering Society
AP	– Access Point
API	– Application Programming Interface
ASCII	– American Standard Code for Information Interchange
AV	– Audio-video, nebo také audiovizuální
CPL	– Composition Playlist
DALI	– Digital Addressable Lighting Interface
DCI	– Digital Cinema Initiatives
DCP	– Digital Cinema Package
DLP	– Digital Light Processing
eSATA	– external Serial Advanced Technology Attachment
IDE	– Integrated Development Environment
IOP	– Interop
IoT	– Internet of Things
ISO/OSI	– International Organization for Standardization / Open System Interconnection
HTTP	– Hypertext Transfer Protocol
HTTPS	– Hypertext Transfer Protocol Secure
GSM	– Groupe Spécial Mobile
KDM	– Key Delivery Message
JNIOR	– Java Network Input Output Resource
JSON	– JavaScript Object Notation
MITM	– Men In The Middle
MVC	– Model View Controller
MXF	– Material Exchange Format
NFC	– Near Field Communication
OV	– Original Version
PCM	– Pulse-code modulation
PHP	– Preprocessor Hypertext
PKL	– Packing List
RTC	– Real Time Clock
SMPTE	– Society of Motion Picture and Television Engineers
SMS	– Short Message Service
SPDIF	– Sony/Philips Digital InterFace
SSL	– Secure Sockets Layer

SQL	– Structured Query Language
SXDR	– Silicon X-tal Reflective Display
TCP/IP	– Transmission Control Protocol/Internet Protocol
TMS	– Theatre Management System
URI	– Uniform Resource Identifier
URL	– Uniform Resource Locator
USB	– Universal Serial Bus
VF	– Verified version
VGA	– Video Graphics Array
VNC	– Virtual Network Computing
XML	– Extensible Markup Language

Seznam obrázků

1	Snímek obrazovky z přehledu DCP v TMS MELODY, zdroj: autor	17
2	Snímek obrazovky z přehledu KDM v TMS MELODY, zdroj: autor	18
3	Snímek obrazovky z přehledu DCP a KDM v kino-serveru GDC SX-3000, zdroj: autor	19
4	Projektor BARCO DP2K-10S, zdroj: barco.com	20
5	Snímek obrazovky webového rozhraní projektoru, zdroj: autor	21
6	Zásuvný kino-server s integrovaným media-blokem GDC SX-3000, zdroj: gdc- tech.com	22
7	Snímek obrazovky ovládacího rozhraní kino-serveru, zdroj: autor	22
8	JNIOR automation controler, zdroj: integpg.com/jnior/	27
9	Snímek obrazovky z vizualizační aplikace appCUE, zdroj: autor	28
10	Jednotka řídicího systému eCNA-10, zdroj: public.eprad.com	29
11	Snímek obrazovky vizualizačního webového serveru eCNA-10, zdroj: film-tech.com	30
12	Snímek obrazovky systému pro správu obsahu Melody TMS, zdroj: cinemanext.com/melody- tms-cinemanext	31
13	Ukázka systému GDC TMS-2000 na různých platformách, zdroj: gdc-tech.com .	32
14	Popis názvu DCP balíku, zdroj: autor	37
15	Systém KDM ochrany, zdroj: cinepedia.com	40
16	E-R diagram databáze a datový slovník, zdroj: autor	44
17	Data-flow diagram MVC aplikace, zdroj: itnetwork.cz	47
18	Aktivity diagram znázorňující spouštění aplikace a přihlášení uživatele, zdroj: autor	60
19	Snímek obrazovky s dialogovým oknem pro přihlášení, zdroj: autor	60
20	Aktivity diagram znázorňující návaznost dalších aktivit dle výběru akce, zdroj: autor	61
21	Snímek obrazovky aktivity ManageDatabaseActivity s fragmentem DcpFragment, zdroj: autor	62
22	Dialogové okno s výpisem KDM přiřazených k danému DCP, zdroj: autor	63
23	Dialogové okno s výpisem vlastností daného DCP, zdroj: autor	63
24	Aktivita pro úpravu stávajícího DCP, zdroj: autor	64
25	Snímek obrazovky aktivity ManageDatabaseActivity s fragmentem KdmFrag- ment, zdroj: autor	65
26	Dialogové okno s výpisem vlastností daného KDM, zdroj: autor	66
27	Aktivita pro úpravu stávajícího DCP, zdroj: autor	67
28	Fragment pro ovládání instalované zvukové jednotky DOLBY CP650, zdroj: autor	68

Seznam tabulek

1	Srovnání základních vlastností relačních databázových systémů	41
2	Srovnání rychlosti odezvy databázového serveru s připojením prostřednictvím mobilních dat	74
3	Srovnání rychlosti odezvy databázového serveru s připojením prostřednictvím telefonní linky CETIN	74

Seznam výpisů zdrojového kódu

1	HTTP požadavek zaslaný metodou GET	34
2	HTTP požadavek zaslaný metodou POST	34
3	Příklad JSON objektu	35
4	Konfigurace routování URL v souboru RouterFactory.php	47
5	Zpracování požadavku v souboru MainPresenter.php	48
6	Získání připojení k databázi v modelu DatabaseManager.php	49
7	Konfigurace připojení k databázi v souboru config.neon	50
8	Vložení a úprava DCP v modelu DcpManager.php	50
9	Odstranění DCP v modelu DcpManager.php	51
10	Odstranění DCP včetně přiřazených KDM v modelu DcpManager.php	51
11	Výběr a seřazení DCP v modelu DcpManager.php	52
12	SQL příkaz pro výběr všech DCP z tabulky dcp seřazených sestupně podle názvu CPL	52
13	Formátování dat a jejich navrácení do presenteru v modelu DcpManager.php . .	53
14	Vložení a úprava KDM v modelu KdmManager.php	53
15	Odstranění KDM v modelu KdmManager.php	54
16	Výběr a seřazení KDM v modelu KdmManager.php	54
17	SQL příkaz pro výběr všech KDM z tabulky kdm seřazených sestupně podle názvu CPL	55
18	Formátování dat a jejich navrácení do presenteru v modelu KdmManager.php . .	56
19	Výběr uživatelů a jejich navrácení do presenteru v modelu UserManager.php . .	57
20	Autentifikace uživatele a ověření hesla v databázi v modelu UserManager.php . .	57
21	Příklad JSON řetězce pro vložení nového DCP	69
22	Příklad JSON řetězce navraceného serverem obsahující více záznamu KDM . . .	69
23	Příklad JSON řetězce obsaženého v požadavku a odpovědi pro nastavení hlasitosti zvukové jednotky na DCI úroveň 4.0 (cca -8.33 dBC)	70
24	Příklad JSON řetězce obsaženého v požadavku pro zjištění všech parametrů zvukové jednotky	70
25	Příklad JSON řetězce obsaženého v odpovědi pro zjištění všech parametrů zvukové jednotky	70
26	Implementace odeslání a zpracování JSONObjectRequest požadavku v souboru MyHttpAdapter.java	71
27	Implementace výběru DCP z databáze v souboru MysqlQuery.java	73

Úvod

Od roku 2013 probíhala v České republice masová vlna digitalizace na standard moderní digitální projekce DCI [23]. Dělo se tak jednak v důsledku již dosluhující analogové technologie filmového pásu šíře 35mm, a také značným tlakem velkých zahraničních producentů a distributorů, pro které již nebylo rentabilní vydávat své digitálně natočené filmy na analogových nosičích, jenž jsou kvalitativně nestálé a na výrobu drahé. Tímto zásahem se podstatně zúžil výběr kin, ve kterých bylo možné zhlédnout premiéry českých i zahraničních filmů. Za kulturou bylo nutné dojíždět do již zdigitalizovaných kin nebo nových multiplexů. [1]

Provozovatelé jednosálových kin v České republice (v drtivém počtu případů přímo nebo nepřímě obce) museli pro udržení diváků před odlivem ke konkurenci a zachování pestré kulturní nabídky chtít nechtít digitalizovat. Cena digitalizace se odvíjela případ od případu a nemalou finanční část tvořila i stavebně technická úprava objektů. [2]

Výrobci digitálních projektorů a ostatních zařízení s nimi souvisejícími nabízejí celou řadu produktů. Při digitalizaci je tak velký prostor pro vhodný výběr toho nejlepšího pro danou situaci a díky konkurenci je možné také významně ovlivnit cenu. Všichni prodejci ale cílí na využití své techniky ve vícesálových kinech, kde je hned několik kusů jejich techniky. Nastudujeme-li podrobněji používanou technologii, tak zjistíme, že výrobci osadili svá zařízení různými komunikačními rozhraními, která přímo vybízejí k napojení na řídicí systém. Ty se využívají zejména u vícesálových kin, kde je možné ovládat projekce na všech sálech z jednoho místa. Naopak tato zařízení postrádají uživatelské prvky pro přímé ovládání, které by naopak zase promítači jednosálových kin uvítali.

Jednosálová kina již dle svého názvu využívají jeden projektor v sestavě tzv. stand-alone. Jako samostatné funkční jednotky jsou v kinech většinou instalovány digitální projektory s média blokem, lokálním úložištěm a zvukovým procesorem. Taková sestava je výbavově identická s tou, která je v multiplexech umístěná na jeden sál a plně odpovídá výše zmíněnému standardu. Obsluha oproti multiplexu postrádá komfort "řídicího centra", které je pro jednu sestavu ovládanou v místě instalace bezpředmětné. Takový systém (TMS) představuje investici minimálně ve výši ceny samotného projektoru, kterou si provozovatelé s ohledem na povahu provozu nemohou dovolit. Uživatel (promítač) tak postrádá uživatelsky přívětivý přehled o stavu zařízení, průběhu projekce, servisních intervalech a plno dalších užitečných informací. Úkony, které provádí pro odbavení projekce se častokrát opakují.

Projekční zařízení je samozřejmě i bez takového systému plně funkční. Jen jeho ovládání je poněkud složitější a veškeré funkcionality nejsou na první pohled zřejmé. Výrobci zkrátka nemyslí na využití v jednosálových kinech, protože se v současné době jedná v západním světě o trend již zcela ojedinělý. [3] V USA převládají od 60. let 20. století multiplexy nebo jednosálová kina jedné značky.

Cílem práce je provést analýzu problému na vzorovém jednosálovém kině a pokusit se navrhnout řešení, které si neklade za cíl nahradit plnohodnotný řídicí systém (TMS) vlastním

řešením, ale pokusit se nalezené řešení aplikovat na co nejvíce provozů s podobnými požadavky, kterým tím ulehčí opakovanou práci a levnějším a zároveň individuálnějším způsobem zpříjemní a zautomatizuje proces digitální projekce.

1 Současný stav

Pro testovací účel bylo vybráno *Kino Orion* v Hradci nad Moravicí. Toto kino je jedno z nejdéle promítajících kin ve Slezsku. Svůj provoz zahájilo v roce 1927. Během druhé světové války byla technika převezena do nedalekého Hlučína. Po válce se provoz obnovil. V 70. letech 20. století byla v kinosále vybudovaná elevace, provedena kompletní rekonstrukce sálu a předsálí pro nový širokoúhlý standard *Cinemascope* a promítací kabina osazena novými stroji MEO5 výroby *Meopta Přerov*. Tyto stroje sloužily jako primární zdroj obrazu až do konce roku 2012. Mezi lety 2013 až 2015 se střídavě promítalo ze staré a dosluhující 35mm technologie a zároveň z nově pořízeného dataprojektoru a Blue-Ray přehrávače (standard E-cinema). V září 2015 byla dokončena rekonstrukce promítací kabiny a kino bylo zdigitalizováno dle standardu DCI. Firmou *KES* byl nainstalován digitální kino-projektor DP2K-10S belgické společnosti *BARCO* s integrovaným media-blokem firmy *GDC SX-3000*. Zvukový řetězec v konfiguraci 7.1 je řízen audio procesorem CP650 společnosti *Dolby Laboratories*. Zachována byla rovněž klasická analogová projekční technologie. Pro filmové pásy širší 35mm byl instalován digitálně řízený projektor MEO5XB3, a pro filmový pás širší 16mm projektor Meopton2. [4]

1.1 Správa digitálního obsahu

Podobně jako u klasické analogové filmové kopie se v digitálních kinech manipuluje i s tou digitální. Dle objednávky a dopravních dispozic distributor vyslal klasickou kopii poštou nebo dráhou do kina, které ji má v programu nasazenou. Kino po jejím odehrání odesílá kopii do dalšího kina nebo zpátky do skladu distributora dle předem stanoveného plánu (dopravní dispozice). [5]

S digitálními kopiemi lze obeslat ve stejný časový úsek podstatně větší počet kin než s kopií klasickou. Digitální balíček (DCP) je distribuován na klasickém pevném disku s rozhraním USB nebo eSATA. Promítačům poté stačí obsah disku zkopírovat (naingestovat) do lokálního úložiště a disk poslat dle dispozic dále. Takový proces je zvládnutelný během několika hodin a další den již může mít disk k dispozici jiné kino.

Digitální kopie přináší samozřejmě své výhody i nevýhody. Pomineme-li kvalitu a další technické aspekty, pak bychom mohli mezi hlavní výhody zařadit možnost nasazení premiérových titulů v prvních dnech jejich uvedení ve více kinech současně při minimálním růstu nákladů a jednoduchá a bezztrátová duplikace kopie.

Mezi nevýhody patří zejména riziko zneužití audiovizuálního díla. Digitální obsah je jednoduše duplikovatelný, a proto distributor odpovědný za dodržování autorských práv nemá zcela kontrolovatelný přehled o počtu kopií. Distributor má seznam kin, které s kopií disponují, ale netuší, jestli ji ihned po odehrání ze svého úložiště smazaly. Kino může teoreticky provést nenahlasenou neveřejnou projekci nebo poskytnout kopii jinému kinu. Tím je samozřejmě distributor a ve výsledku i producent díla krácen na zisku a vzniká mu majetková újma.

U atraktivnějších titulů velkých zahraničních ale i tuzemských produkcí se využívá takzvaného KDM managementu. Výrobce digitální kopie podmíní hratelnost (nikoliv dispozici) digitální kopie platným klíčem (KDM). Klíče jsou poté vystavovány dle objednávek přímo jednotlivým kinům. Malý XML soubor promítač stáhne například z e-mailu a naingestuje do media bloku. Tím se klíč spojí s patřičným DCP a to se stane po dobu specifikovanou vystavovatelem klíče hratelné. Takto je zamezeno nechtěnému úniku audiovizuálního obsahu, protože je KDM jednak pevně spojeno s konkrétní jazykovou a prostorovou verzí DCP, a jednak s unikátním sériovým číslem kino-serveru.

Distributoři tím dostávají jasnou záruku nezneužití díla, ovšem pro provozovatele a promítače roste další administrativa a činnost související s kompletací DCP obsahu a jeho následnou projekcí. Pro provoz kina je nutné vést sofistikovanou evidenci o DCP a KDM, které již dorazily od distributorů, respektive předchozích kin v dopravním řetězci. Problematiku dále rozšiřuje více jazykových verzí některého titulu nebo i různé obrazové či prostorové verze (2D, 3D, DOLBY ATMOS, DBOX).

Ve vícesálových kinech (s více projektory) se o ingest a správu klíčů stará TMS. TMS je v podstatě server s velkým datovým úložištěm a speciálním softwarem výrobce media bloku (nebo kino-serveru). Promítač tak ingestuje DCP a KDM pro každý media blok do centrálního serveru. Z nich si poté posílá po lokální síti obsah do jednotlivých sálů. Operátor TMS má okamžitý a uživatelsky přívětivý přehled o stavu serveru, platnosti klíčů atp. Rovněž je možné zobrazit si sestavu titulů, které má již nahrány nebo odemčeny na nejbližší dobu, či které chybí a je třeba kontrolovat jejich včasné doručení. Případně může po vhodném nastavení samotná TMS provádět hlídání a vedoucího projekce o nedostacích informovat pomocí e-mailu či SMS. [6]

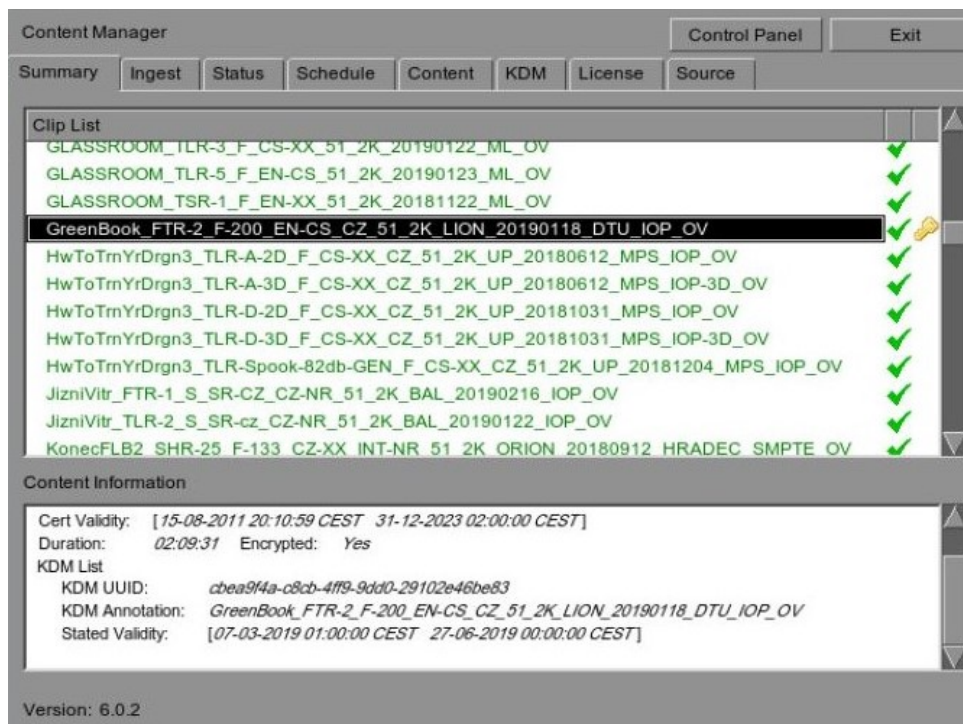
Name	KDM	2D/3D	OV/VF	Duration	Size	Added
3DByte_upozorneni_2D_Disney_Flat	Not required	2D	OV	00:00:05	0 MB	24/04/2018
A-COMPANY-CZECH_SHR_F-XX-XX_CZ_51_2K_20131206_UTD_OV	Not required	2D	OV	00:00:14	0 MB	25/04/2018
Alpha_FTR-1-2D_S-EN-CS_CZ_51-Dbox_4K_SPE_201	Unavailable	2D	OV	01:36:25	79 GB	04/12/2018
Alpha_FTR-1-2D_S-EN-XX_OV_51-Dbox_4K_SPE_201	Unavailable	2D	OV	01:36:25	79 GB	04/12/2018
Aquaman_FTR-2D_S-CS-XX_CZ_51_2K_WR_201811	Available	2D	OV	02:23:09	0 MB	06/12/2018
Aquaman_FTR-3D-45R_S-CS-XX_CZ_51_2K_WR_201	Available	3D	OV	02:23:09	0 MB	06/12/2018
Aquaman_FTR-3D-45R_S-EN-XX_OV_51_2K_WR_20181114_DTB_IOP-3D_OV	Unavailable	3D	OV	02:23:09	0 MB	06/12/2018
Aquaman_FTR_S-EN-CS_CZ_51_2K_WR_20181127_DTU_IOP_VF	Available	2D	OV	02:23:09	0 MB	06/12/2018
Aquaman_FTR_S-EN-XX_OV_51_2K_WR_20181103_DTB_IOP_OV	Unavailable	2D	OV	02:23:09	0 MB	06/12/2018
ASimpleFavor_FTR-2-F-EN-CS_CZ_51_2K_LION_20180903_DTU_IOP_OV	Available	2D	OV	01:56:42	0 MB	17/10/2018
Assassination_FTR-3_S-EN-CS_CZ_51_2K_UP_20181101_MPS_IOP_VF	Unavailable	2D	VF	01:48:15	0 MB	20/11/2018
Assassination_FTR-3_S-EN-XX_INT_51_2K_UP_20180825_MPS_IOP_OV	Unavailable	2D	OV	01:48:15	135 GB	09/11/2018
AStarIsBorn_FTR-4_S-EN-CS_CZ_71_2K_WR_20180913_DTU_IOP_VF	Unavailable	2D	OV	02:15:37	0 MB	25/09/2018
ASTERIXaTajemstvíLehtivaru_FTR1_S-CS-XX_CZ_51_2K_20181208_CNK_IOP_OV	Not required	2D	OV	01:25:19	63 GB	17/12/2018
AvengersinWar_FTR-1-2D_S-CS-XX_CZ_51-Dbox_2K_DL_20180415_DTU_IOP_VF	Unavailable	2D	OV	02:29:45	0 MB	24/04/2018

Obrázek 1: Snímek obrazovky z přehledu DCP v TMS MELODY, zdroj: autor

Name	Start	End	Status	Ingested	Booth
Aquaman_FTR_S_EN-CS_CZ_51_2K_WR_20181127_DTU_IOP_VF	12/13/18	3/13/19	Valid	Yes	2
Aquaman_FTR_S_EN-CS_CZ_51_2K_WR_20181127_DTU_IOP_VF	12/13/18	3/13/19	Valid	Yes	4
Aquaman_FTR_S_EN-CS_CZ_51_2K_WR_20181127_DTU_IOP_VF	12/13/18	3/13/19	Valid	Yes	6
Aquaman_FTR_S_EN-CS_CZ_51_2K_WR_20181127_DTU_IOP_VF	12/13/18	3/13/19	Valid	Yes	1
Aquaman_FTR_S_EN-CS_CZ_51_2K_WR_20181127_DTU_IOP_VF	12/13/18	3/13/19	Valid	Yes	5
Aquaman_FTR_S_EN-CS_CZ_51_2K_WR_20181127_DTU_IOP_VF	12/13/18	3/13/19	Valid	Yes	3
Aquaman_FTR-2D_S-CS-XX_CZ_51_2K_WR_20181128_DTU_IOP_VF	12/13/18	3/13/19	Valid	Yes	2
Aquaman_FTR-2D_S-CS-XX_CZ_51_2K_WR_20181128_DTU_IOP_VF	12/13/18	3/13/19	Valid	Yes	4
Aquaman_FTR-2D_S-CS-XX_CZ_51_2K_WR_20181128_DTU_IOP_VF	12/13/18	3/13/19	Valid	Yes	6
Aquaman_FTR-2D_S-CS-XX_CZ_51_2K_WR_20181128_DTU_IOP_VF	12/13/18	3/13/19	Valid	Yes	1
Aquaman_FTR-2D_S-CS-XX_CZ_51_2K_WR_20181128_DTU_IOP_VF	12/13/18	3/13/19	Valid	Yes	5
Aquaman_FTR-2D_S-CS-XX_CZ_51_2K_WR_20181128_DTU_IOP_VF	12/13/18	3/13/19	Valid	Yes	3
Aquaman_FTR-3D-45R_S-CS-XX_CZ_51_2K_WR_20181129_DTU_IOP-3D_VF	12/13/18	3/13/19	Valid	Yes	2
Aquaman_FTR-3D-45R_S-CS-XX_CZ_51_2K_WR_20181129_DTU_IOP-3D_VF	12/13/18	3/13/19	Valid	Yes	4
Aquaman_FTR-3D-45R_S-CS-XX_CZ_51_2K_WR_20181129_DTU_IOP-3D_VF	12/13/18	3/13/19	Valid	Yes	6
Aquaman_FTR-3D-45R_S-CS-XX_CZ_51_2K_WR_20181129_DTU_IOP-3D_VF	12/13/18	3/13/19	Valid	Yes	1

Obrázek 2: Snímek obrazovky z přehledu KDM v TMS MELODY, zdroj: autor

V jednosálových kinech (s jedním digitálním projektorem) mají promítači společně s absencí TMS také absenci vhodně zobrazeného přehledu. Pro kontrolu již naingestovaných a odemčených titulů tak musí spustit celou projekční soustavu. Ručně vyhledat jednotlivé tituly a ověřit jejich platnost. Absenci těchto systémů řeší jednotlivá kina různě a možností se nabízí hned několik. Od pověření odpovědností za kompletnost nasazených titulů jedním pracovníkem (např. vedoucí projekce) až po hromadné e-mailové korespondence všem zaměstnancům. Kteroukoliv z výše popsaných metod můžeme označit za plýtvání pracovního fondu zaměstnanců, kteří pravidelně provádějí stejnou činnost, jež se dá zautomatizovat a zjednodušit. Evidence prováděná pomocí sdílených tabulek neumožňuje využít pokročilé programátorské metody a taková "aplikace" je zcela prostá například ošetření vstupů. Uživatel tak může nevědomky zadat nesprávné údaje, čímž vznikne chyba lidského faktoru. Na základě těchto údajů se za pomoci dalších uživatelů může zřetězovat chyba, kterou se nemusí podařit ihned odhalit.



Obrázek 3: Snímek obrazovky z přehledu DCP a KDM v kino-serveru GDC SX-3000, zdroj: autor

1.2 Vzdálené ovládání projekce

První pokusy o automatizaci projekce se projevovaly již na klasickém filmovém pásu. Filmové laboratoře nalepily mimo obrazovou a zvukovou část (nejčastěji na okraj) značky v podobě fólie z magnetického materiálu. Značky byly charakteristické svým umístěním, délkou nebo magnetickou polarizací. Byl-li projektor vybaven čtecím zařízením magnetických značek a jeho instrukční signálové linky správně propojeny, pak projektor po přečtení startovací značky roztočil motory druhého stroje s již založeným dalším dílem filmu a po přečtení prolínací značky (typicky po 5-ti až 7-mi sekundách) prolnul obraz mezi díly, čímž plynule navázal na předchozí díl. Odpadla tím starost promítače, který většinou nebyl natolik pohotový a přepnutí strojů mohlo být i méně všímavým divákem rozpoznáno. Později byly vyvinuty a používány značky na rozsvícení světel v sále po skončení projekce nebo značka na synchronizaci externího titulkovacího zařízení. [5]

1.2.1 Vzdálené ovládání a komunikační rozhraní digitálního projektoru

Digitální projektor splňující standard DCI se od klasického dataprojektoru liší. DCI projektor sám o sobě je pouze šasi s objektivem, lampou a základní elektronikou pro pohyb objektivu, obsluhu ventilátorů či teplotních čidel. Dle normy DCI [23] musí být soustava doplněna o zobrazovací technologii a media-blok napojený na kino-server sloužící k odbavení DCP obsahu. Výrobci projektorů proto do svých zařízení umísťují zásuvné sloty, do kterých se vloží media-

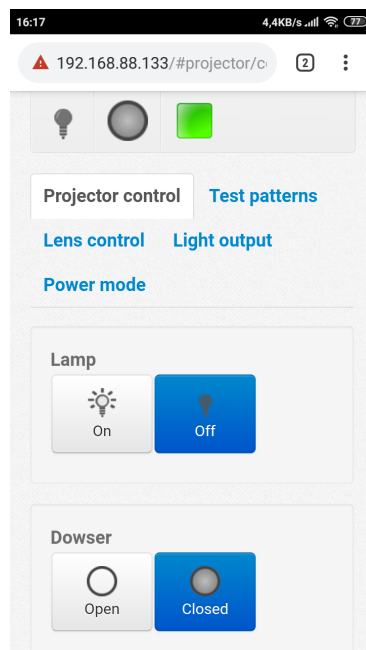
blok třeba i jiného výrobce a zobrazovací zařízení výrobce technologie (DLP, SXDR). Vše musí být kompatibilní dle normy DCI, která tímto rozdělením určuje výrobcům techniky i obsahu jasný rozsah jejich povinností a zároveň i odpovědností za kvalitu projekce. [7]

Projektory je možné ovládat přímo, pomocí hardwarově vestavěných tlačítek, kterými se aktivují základní funkce projektoru, jako je ovládání lampy, klapky, předvolených maker, polohy objektivu atd. V bezprostřední blízkosti projektoru se nachází také dotykový panel. Z něj lze již plně ovládat projektor (výjimečné i media-blok), včetně získání podrobných informací o stavu stroje nebo provedení základních i pokročilých servisních zásahů. K projektorům bývá dodávána i počítačová aplikace, která plně kopíruje vzhled i funkcionalitu dotykového panelu. [9]

V současném stavu je možné instalovaný projektor v *Kině Orion Barco* DP2K-10S ovládat dotykovým panelem nebo vzdáleně přes počítačovou aplikaci. Další možností vzdáleného ovládání je webové rozhraní. Po zadání přihlašovacích údajů se uživateli nabízí plně responzivní webová stránka, jež dobře vypadá na různých velkých mobilních zařízeních. Vývoj čehokoliv pro vzdálené ovládání se po následujícím popisu může zdát jako zbytečný, protože již od pohledu je zřejmé, že vzdálená správa je zde dopodrobna vyřešena. Samotný projektor je avšak pouze zprostředkovatelem vizuálního požitku a k jeho obsluze přistupuje promítač pouze při startu nebo ukončení projekce. S tímto typem projektoru lze komunikovat také pomocí TCP/IP protokolu, stejně jako počítačová aplikace. Set příkazů je dán k dispozici výrobcem. [8]



Obrázek 4: Projektor BARCO DP2K-10S, zdroj: barco.com



Obrázek 5: Snímek obrazovky webového rozhraní projektoru, zdroj: autor

1.2.2 Vzdálené ovládání a komunikační rozhraní kino-serveru

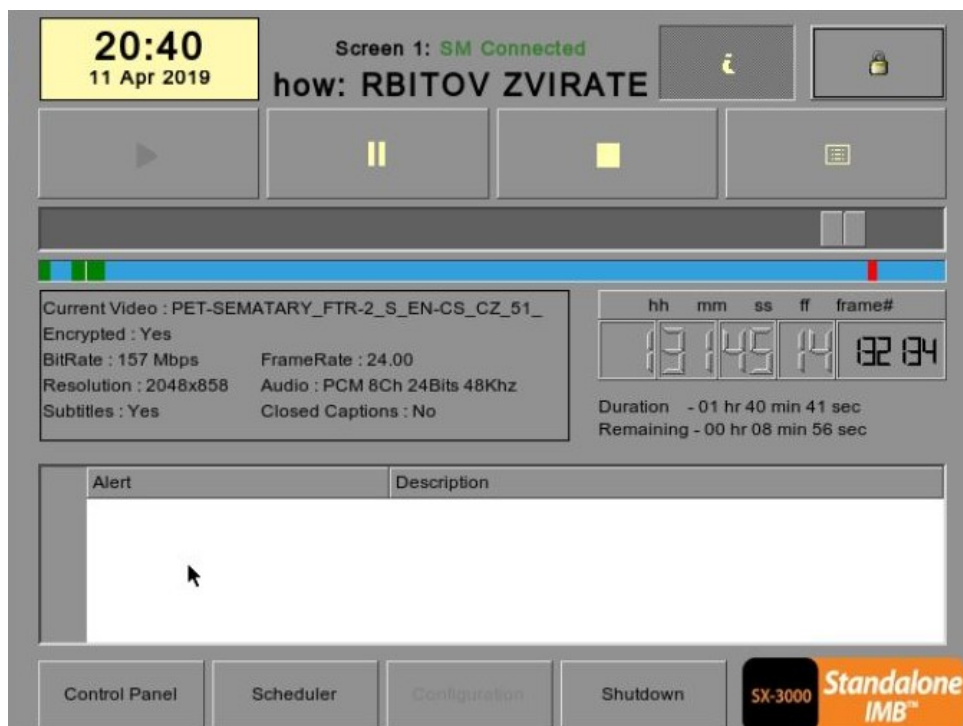
Kino-server je zařízení, které se skládá z počítače s operačním systémem založeným povětšinou na linuxové platformě, diskového pole (jako externí zařízení) a samotného media-bloku. Jedná se o velmi výkonnou grafickou kartu, která zpracovává vstupní audio a video soubory. Zvukové předává na digitální vstup zvukového procesoru, obrazové posílá přes paralelní sběrnici na řídicí desku zobrazovacího čipu. Ta vystavuje v reálném čase obrazový soubor na jednotlivé pixely čipu.

Ovládání samotného kino-serveru je pro promítače stěžejní. Do jeho úložiště nahrává DCP soubory a sestavuje zde playlist (show), kde určuje jaké upoutávky a reklamy se před filmem nasadí. Do show poté dle potřeby vkládá automatizační značky. Ty dají pokyn projektoru, aby přezoomoval objektiv při změně obrazového formátu nebo zesílil či zeslabil zvuk. K ovládání může sloužit kupříkladu dotykový panel projektoru, pokud je výrobce projektoru a kino-serveru shodný. [13]

Ovládání instalovaného kino-serveru s integrovaným média-blokem SX-3000 od firmy *GDC* je možné pomocí monitoru připojeného přes rozhraní VGA přímo do kino-serveru a klasické myši nebo klávesnice přes USB. Vzdálené ovládání je možné pomocí VNC připojení. Komunikace s tímto zařízením je možná také přes TCP/IP protokol realizovaný pomocí metalické sítě Ethernet. Sestavu příkazů je možné si nadefinovat v nastavení serveru dle svých potřeb a požadavků. [10]



Obrázek 6: Zásuvný kino-server s integrovaným media-blokem GDC SX-3000, zdroj: gdc-tech.com



Obrázek 7: Snímek obrazovky ovládacího rozhraní kino-serveru, zdroj: autor

1.2.3 Vzdálené ovládání a komunikační rozhraní zvukového procesoru

Zvukový procesor je zařízení, které přímo zpracovává různé formy analogových či digitálních audio vstupů a provádí nad nimi předem definované operace v časové a frekvenční oblasti. Signál zesiluje na uživatelem nastavenou hodnotu a posílá na linkové vstupy koncových audio zesilovačů jednotlivých kanálů nebo zvukových okruhů. Zvukový procesor má v sobě nadefinovány

standardsy používané od vzniku zvukové kinematografie, přičemž ty nejpoužívanější formáty má uživatel možnost přepínat pomocí osmi tlačítek. Mezi nejčastější zvukové formáty patří: [11]

- **Mono** (také Academic mono) je monofonní jednostopý signál snímáný z klasického filmového pásu pomocí optického snímače prosvětlený červeným světlem.
- **Dolby A** (také Dolby A-type) je dvoustopý stereo nebo vícekanálový signál z filmového pásu snímáný totožným způsobem jako signál popsáný výše. Oproti monofonnímu signálu zvukový procesor zeslabí oblast vyšších kmitočtů, která je kvůli redukci šumu při pořizování záznamu záměrně zesílena. Každou ze stop čte procesor zároveň i ve fázi posunutou o $+$ nebo $- 90^\circ$, čímž získá další dva zakódované kanály v původních dvou stereofoniích stopách (středový a efektový).
- **Dolby Stereo** je technicky pokročilejší způsob záznamu založený na formátu Dolby A.
- **Dolby Digital** představuje diskrétní záznam signálu reprezentovaný monochromatickými 2D kódy mezi perforací. Nese signál pěti plno rozsahových signálů a jednoho efektového, nízkofrekvenčního.
- **PCM 6-channel** je digitální vstup kompatibilní se standardem AES. Složí k přímému propojení s kino-serverem ve zvukové konfiguraci 5.1.
- **Dolby Surround EX** je oproti předchozímu formátu rozšířený o dva surroundové kanály na konfiguraci 7.1.

Výčet formátů se pochopitelně může lišit dle instalovaného vybavení projekční místnosti. Lze připojit třeba klasický mixážní pult, který směšuje další vstupy nebo Blue-Ray přehrávač pomocí digitálního rozhraní SPDIF.

Promítač potřebuje během projekce regulovat hlasitost. Úroveň hlasitosti má přímý dopad na kvalitní projekci a na podvědomé vnímání uměleckého díla. I když jsou dle standardu DCI míchány snímky na referenční hlasitost a každý kinosál i masterovací studio je dle těchto pravidel změřeno a jeho zvuková aparatura nastavena, existují faktory, které žádné pravidlo nemůže zohlednit. Každý kinosál je jinak utlumen proti odrazům a dozvukům. Ať již záměrně nebo svou nedokonalostí. Rovněž počet diváků v sále je proměnlivý, stejně jako jejich generační složení. Nelze promítat hudební tituly se stejnou hlasitostí pro dospělé a pro seniory. Promítač musí respektovat všechny aspekty a poslechem v sále skutečnou hlasitost regulovat.

Vzdálená regulace hlasitosti a přepínání formátu je u některých výrobců možné provádět opět pomocí VNC připojení realizovaného například pomocí tabletu, který má promítač u sebe v sále nebo pomocí vzdálených drátových či bezdrátových jednotek. [12]

Ve zkoumaném kině je použit zvukový procesor firmy *Dolby Laboratories* s označením CP650. Vzdálené ovládání tohoto procesoru je možné pomocí speciální vzdálené jednotky Remote Control Unit Cat. No. 779 nebo Remote Fader Cat. No. 771. Cena této jednotky se pohybuje od 1000 €

výše. Navíc se jedná o jednotku, jež komunikuje po drátové sběrnici, skrze kterou je také napájena. Ovladač tedy není přenosný, musí být umístěn v sále na fixním místě a jeho použití musí být blokováno proti zneužití neoprávněnými osobami. Další možností, kterou tento procesor nabízí je připojení 100k/N potenciometru. Jedná se o podstatně levnější variantu než výše. Oproti nevýhodám předchozího řešení se použitím potenciometru zcela vyřadí regulace hlasitosti v kabině. Procesor umožňuje také komunikaci pomocí protokolu TCP/IP, kterým komunikuje s projektořem a kino-serverem. Stejný rozsah komunikace lze realizovat i pomocí sériové linky RS-232. [11]

1.2.4 Vzdálené ovládání a komunikační rozhraní 35mm kino-projektoru

Pro zachování možnosti odbavení klasické 35mm analogové projekce pro archivní tituly v původní kvalitě se v kinech zachovávají i klasické projektořy. Jejich ovládání (co se ovládacích prvků týká) je podstatně jednodušší než u digitálních projektořů. S tím je také omezena možnost jejich využití pro další účely a je stanovena výhradně na odbavení analogových kopií.

Obsluha takového stroje musí být řádně zaškolená a projít kurzem promítače nových technologií. Požadavky na promítače jsou zcela jiné než na promítače, který odbavuje digitální projekci. [24] Obsluha musí být pečlivě seznámena s jeho funkcí, stejně tak jako se základními pojmy optiky, elektroniky a mechaniky. Klasický projektoř je totiž více než elektronické zařízení spíše mechanické. Pro zajištění plynulého, stabilního, klidného a ostrého obrazu má promítač k dispozici několik nastavovacích kladek, přitlačných šroubů a dalších mechanických ovladačů.

Z popisu výše vyplývá, že obsluha musí veškeré tyto úkony provádět v projekční kabině a provedenou změnu kontrolovat osobně v sále. Modernější stroje ze začátku tohoto tisíciletí jsou vybaveny motory, které umožňují vzdálené ovládání alespoň těch nejdůležitějších částí elektricky. Ze sálu je pak vhodné ovládat ostření obrazu. Klasická kopie trpí degradací materiálu, který mimo jiné mění i svou tloušťku. Na začátcích a koncích není materiál většinou tolik vysušený jako uprostřed kopie. Proměnlivá tloušťka má poté významný vliv na ostrost obrazu, kterou je nutné korigovat. Také ovládání prolínacích klapek je příjemným usnadněním. V případě, kdy není kopie prolínacími značkami vybavena, je tato práce výhradně na promítači.

V *Kině Orion* je používán stroj MEO5XB3, poslední a nejmodernější model firmy *Meopta Přerov*. Podobné byly instalovány i do celé sítě tehdejších (2008) multiplexů. Tento stroj je vybaven dvěma motory s kluznou spojkou, které umožňují ovládat mechanismus ostření stroje a rámování obrazu. Motoricky je zde řešena také výměna objektivu při změně projekčního formátu. Společně s kontakty pro stykače napájení lampy, hlavních motorů a elektromagnetu prolínací klapky jsou vyvedeny na ovládací svorkovnici. Přivedením řídicího signálu 24V na jednotlivé svorky se daný motor začne otáčet, případně je daný stykač sepnut/rozepnut. Vnitřní logika je řešena pomocí programovatelného automatu SIEMENS LOGO! [14]

1.2.5 Vzdálené ovládání a komunikační rozhraní osvětlovacího systému

Dle platných norem [25] je důležité, aby byl kinosál vybaven několika stupni osvětlení. Během projekce musí být spuštěné nouzové osvětlení, které nerušeně osvětluje hlavní a nouzový východ ze sálu. Nouzové osvětlení musí zůstat provozuschopné i během výpadku elektrické energie. Jedná se o komplexní systém v rámci celé budovy a není nutné jej řešit v rámci této práce. Důležité je dvoustavové ovládání hlavního osvětlení a stmívače pro pomocné osvětlení. Stmívač plynule reguluje intenzitu pomocného osvětlení před spuštěním a po skončení projekce tak, aby se sítnice lidského oka postupně adaptovala na osvětlení okolí. V kinosále je možné pomocí stmívače regulovat několik reflektorů namířených na plátno. Ty se využívají při besedách, uvedení filmů, přednáškách nebo divadelních vystoupeních.

Stmívače nabízené na trhu mohou být řízeny jednak ručně z místa jejich instalace pomocí mechanických ovládacích prvků nebo vzdáleně. Kromě ethernetu nabízejí také sběrnici RS-232 nebo unifikovanou světelnou sběrnici DMX512.

Touto sběrnici jsou vybaveny také regulátory umístěné ve zkoumaném kině. Bohužel se nenacházejí na stejném místě a hromadné ovládání všech regulátorů je možné pouze ze sálu. Z projekční kabiny je část osvětlení neovladatelná.

1.2.6 Vzdálené ovládání a komunikace s ostatními zařízeními

V kinech mohou být instalována i další zařízení, vše souvisí s původním určením kina a jeho vybavením.

Promítací kabina *Kina Orion* je vybavena odtahem teplého vzduchu produkovaného během provozu projekčních lamp. Krom chlazení a cirkulace vzduchu uvnitř lampové skříně odjímá z kabiny také ozon, který provozem xenonových výbojek vzniká. Motoricky je také řešeno maskování plátna černými oponami.

Ovládání ostatních periférií, jako je třeba výše zmíněný odtah nebo oponový systém, vzdáleně jen zvýší uživatelský komfort. Ke každému zařízení je ale třeba přistupovat individuálně, dle jeho možností a požadovaných potřeb.

1.3 Požadavky na aplikaci

Z výše uvedeného popisu současného stavu je patrné, že chybí jednotný systém komunikace se všemi perifériemi. Nabízí se řešení centrálního pultu, kde by měl operátor přehled o všech zařízeních pohromadě v podobě přehledné vizualizace. Hned první nevýhodou takového pultu je jeho případná velikost a imobilita. V případě, že neprobíhá klasická filmová projekce, ale kupříkladu vzdělávací přednáška, je nutné, aby si přednášející, případně jeho asistent, mohl zařízení ovládat sám dle svých potřeb. Nejeví se jako profesionální, aby přednášející seděl na boku nebo v zadní části sálu během své prezentace nebo z místa před plátnem dával na dálku pokyny svému asistentovi.

Než se pustit do vývoje vlastního ovládacího panelu je daleko výhodnější použít dostupné mobilní zařízení s operačním systémem. Dnes již každý tablet nebo mobilní telefon disponuje komfortní dotykovou obrazovkou a bezdrátovou komunikací. Díky zabezpečené lokální síti je každé zařízení z předchozí podkapitoly v podstatě dostupné jinému, včetně ovládacího tabletu skrze Wi-Fi připojení. Každé zařízení komunikuje pomocí různých protokolů realizovaných pouze na stejné fyzické vrstvě. Z tohoto důvodu bylo rozhodnuto pro řešení komunikace s centrální řídicí jednotkou pomocí HTTP požadavků, která je bude zpracovávat a pomocí svých distribuovaných periférií ovládat jednotlivá zařízení. Tato centrální řídicí jednotka zprostředkuje rovněž komunikaci s perifériemi, které nekomunikují pomocí sítě. Pro tyto periférie by bylo použití takového zařízení nevyhnutelné.

V rámci jedné aplikace pro mobilní platformy bylo rozhodnuto vyřešit i evidenci DCP obsahu a KDM klíčů pro otevření projekce. Tato evidence bude založena na relačním databázovém systému, který nebude umístěn na stejném zařízení. Nevýhodou je nutnost připojení k databázi. Výhodou je poté možnost napojení více instalací takové aplikace na různých zařízeních na jednu databázi běžící na vzdáleném serveru. Změna záznamu jedním uživatelem je tím pádem okamžitě přístupná i pro ostatní. Separovaný databázový server umožňuje jednodušší napojení na případnou webovou nebo jinou podobu další aplikace.

2 Systémy pro automatizaci a správu obsahu

Na trhu je celá řada profesionálních systémů pracujících na podobné bázi jako aplikace, která je výsledkem této bakalářské práce. Nicméně žádný ze systémů nekombinuje funkci bezdrátového ovládání a správy digitálního obsahu do jednoho ani přímou interakci mezi nimi.

2.1 Přehled systému pro automatizaci

Systémy pro automatizaci jsou založeny na řídicích jednotkách, které obsahují rozhraní, sběrnice a rozšiřující periferie, skrze které se připojují jednotlivá zařízení. Samotná vizualizace řízení je řešena operátorskými panely, dotykovými displeji nebo webovými aplikacemi

2.1.1 JNIOR automation controller

Jedním z kontrolérů, který je možné napojit na projekční technologii je JNIOR automation controller od firmy *INTEG*. Název produktu JNIOR je zkratka pro Java Network Input Output Resource.

JNIOR obsahuje osm vysoko impedančních vstupů a osm reléových výstupů, kdy každé relé má maximální spínací schopnost 24 V DC / 1 A. Relé mají typ kontaktů normally open, přičemž dvě z nich jsou konfigurovatelné jako normally closed. Dále kontrolér obsahuje dvě sériové linky RS-235 a ethernetový port. K provozu zařízení je třeba externí napájecí zdroj 12 - 24 V DC/1 A. [15]



Obrázek 8: JNIOR automation controller, zdroj: integpg.com/jnior/

Na procesoru, který řídí periferie běží operační systém Linux. Díky tomu je možné spouštět různé procesy a aplikace, které mají přístup k hardwarovým vstupům a výstupům. Toho využívají i výrobci projekční techniky, kdy vytvořili program pro komunikaci s jejich zařízením. To se pomocí ethernetu nebo sériové linky propojí s kontrolérem, kde se spustí vytvořený program. V zařízení komunikujícím s kontrolérem si poté pomocí grafického uživatelského rozhraní

integrátor navolí jednotlivé události, které napojí na další navazující akce (spuštění projekce, zapnutí ventilace). Celý proces implementace se objede bez jediného řádku kódu. [10]

Největší nevýhodou je, že JNIOR postrádá jakoukoliv podobu vizualizace. Ta se řeší externími systémy s programovým napojením a celá implementace se stává složitější. Základní konfigurace tohoto systému ve své podstatě pouze rozšiřuje komunikační rozhraní jednotlivých zařízení. Řídicí kontrolér JNIOR se vyrovná dnešním jednodeskovým počítačům jako je Raspberry Pi, Beagle Bone Black nebo Bannna Pi rozšířený o silové vstupy a výstupy. [15]

2.1.2 CUE controlCUE-one

Řízení a vizualizace je založená na systému vyvinutém českou firmou *CUE, a.s.* Řízení obstarává hlavní jednotka, která je pomocí ethernetu připojena do lokální sítě. Hlavní jednotka zpracovává veškeré příchozí a odchozí požadavky a předává je svým distribuovaným perifériím, se kterými komunikuje pomocí rozhraní RS-485. Periférie obsahují reléové výstupy, binární vstupy, sériové komunikační linky a KNX či DALI bránu pro řízení provozu projekčního sálu.

Vizualizaci obstarává aplikace běžící na platformě iOS, která komunikuje s hlavní jednotkou pomocí bezdrátového připojení k síti Wi-Fi.



Obrázek 9: Snímek obrazovky z vizualizační aplikace appCUE, zdroj: autor

Systém byl původně navržen pro využití ve smart home aplikacích, což je také hlavní oblast působnosti výše zmíněné firmy. Teprve později se přikročilo k jeho použití v oblasti řízení projekční technologie. Implementace se provádí programováním jednotlivých modulů, na které je třeba školení. Většinou celou implementaci zajišťuje jiná firma, než která instaluje digitální technologii. [16]

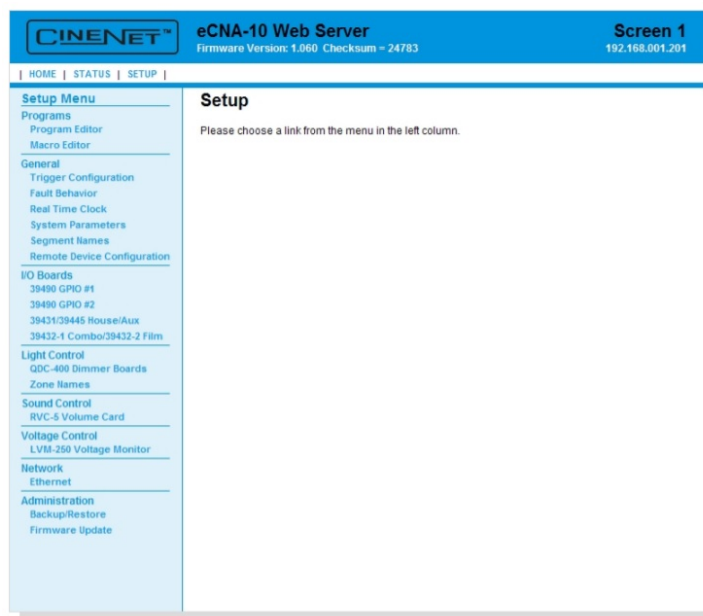
2.1.3 eCNA-10

Řídicí systém eCNA-10 americké společnosti *REGAL ENTERTAINMENT GOUP* byl vyvinut speciálně pro sály dle specifikace DCI. Oproti předchozím systémům byl při vývoji předurčen pro použití v kinech. Jedná se o zařízení určené pro montáž do standardního 19-ti palcového racku. Základní jednotka obsahuje digitální vstupy a výstupy, sériovou komunikační linku RS-232 a ethernet. V případě, že je počet periférií nedostačující, je možné přidat další pomocí rozšiřujících modulů (opět určených pro montáž do racku). Nevýhodou tohoto systému je jeho centralizace. Veškeré zařízení připojené na systém musí své komunikační linky svést do jednoho místa (racku), kde je jednotka potažmo rozšiřující moduly nainstalována. V případě doinstalace ke stávající technologii to může způsobit problémy s dodatečnou kabeláží.[17]



Obrázek 10: Jednotka řídicího systému eCNA-10, zdroj: public.eprad.com

Vizualizace je řešena pomocí webové aplikace běžící na samotné hlavní jednotce. Nutno podotknout, že se jedná o starší systém nasazovaný v kinech zejména na počátku jejich digitalizace. Proto je ovládání webového rozhraní na dotykových obrazovkách neintuitivní. Celý systém byl především navrhnout pro jednorázové naprogramování a bezobslužné řízení sálu. Proto nabízí pokročilé programovací funkcionality pro každé zařízení a postrádá prvky pro okamžité řízení. [17]



Obrázek 11: Snímek obrazovky vizualizačního webového serveru eCNA-10, zdroj: film-tech.com

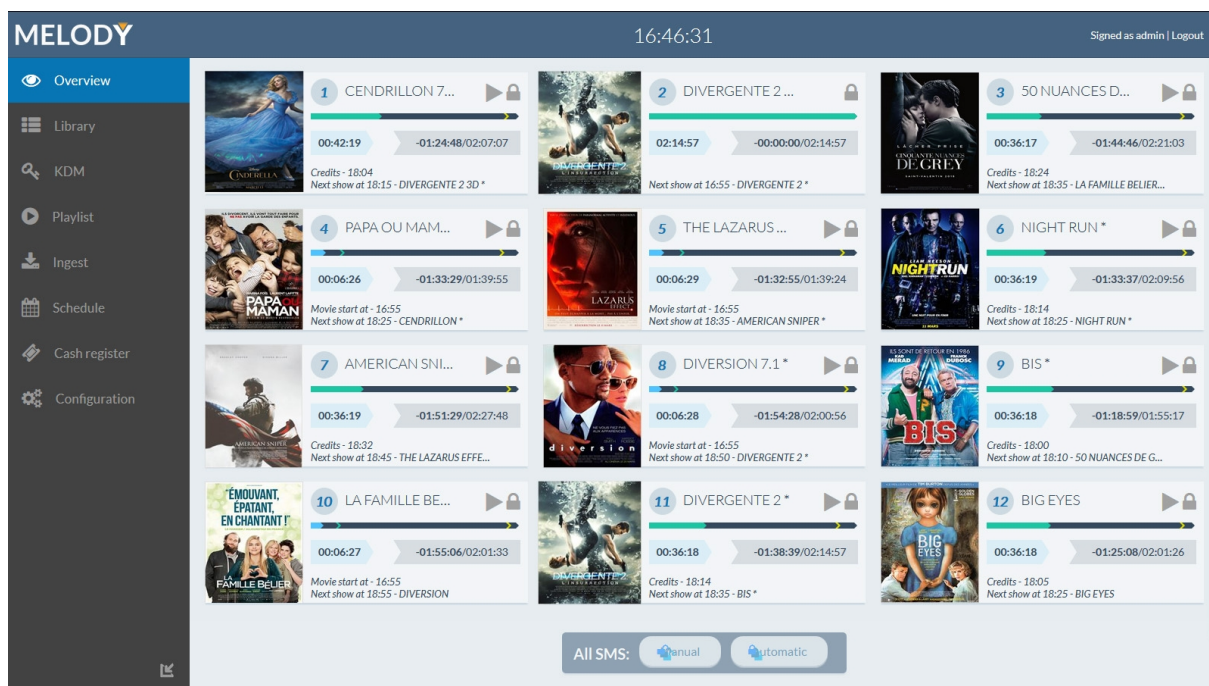
2.2 Přehled systému pro správu obsahu (TMS)

Evidence a správa digitálního obsahu je nedílnou součástí provozu vícesálového kina (multiplexu), kde je bez takového systému provoz v ekonomických mezích neudržitelný. Projekce probíhají v několika sálech paralelně a není v možnostech jednoho zaměstnance všechny projekce hlídat, případně řešit potíže, které se mohou vyskytnout. Aby se usnadnilo hlídání projekcí a jejich odbavení nabízejí výrobci TMS systémy, které celý proces podstatně zjednodušují.

2.2.1 Melody TMS

Systém Melody TMS od belgické firmy *CinemaNext* je kompatibilní s libovolnými zařízeními výrobců nejen projekčních technologií, ale třeba i pokladních systémů. Poskytuje tedy kompletní zastřešení pro provoz více-sálového kina. Z pohledu operátora projekce (promítače) se ovládá z počítače, tabletu nebo mobilního telefonu, který je připojen do lokální sítě stejně jako server s úložištěm, na kterém software Melody TMS běží. Do datového úložiště si promítač stahuje DCP a KDM a také sestavuje show včetně řízení osvětlení, zvuku a ventilace. Takto sestavené show společně s DCP a KDM odešle do konkrétní projekční sestavy pro daný sál. Dále se skrze software naprogramuje automatické spuštění projekce. Díky napojení na pokladní systémy je možné projekci nespouštět v případě, že nedorazí žádný divák zcela automaticky bez zásahu promítače nebo jiného personálu. Pokud je k dispozici i soupis projekcí, dokáže systém automaticky ohlídat dostupnost DCP a platnost KDM. Jako jeden z mála systémů umožňuje i okamžité řízení v případě, kdy je sál přeměněn na konferenční místnost. [18]

Díky kooperaci s různými výrobci je možné instalovanou projekční techniku měnit za novou od jiného výrobce nezávisle na výměně systému Melody TMS za jiný. To je jeho největší výhodou. Díky této vlastnosti je vyšší i jeho pořizovací cena 10 000 € a dalšího náklady spojené s licencemi za provoz. Pro jednosálová kina jsou některé funkcionality zbytečné a pořízení drahého systému je proto odradí.



Obrázek 12: Snímek obrazovky systému pro správu obsahu Melody TMS, zdroj: cinemane.com/melody-tms-cinemane

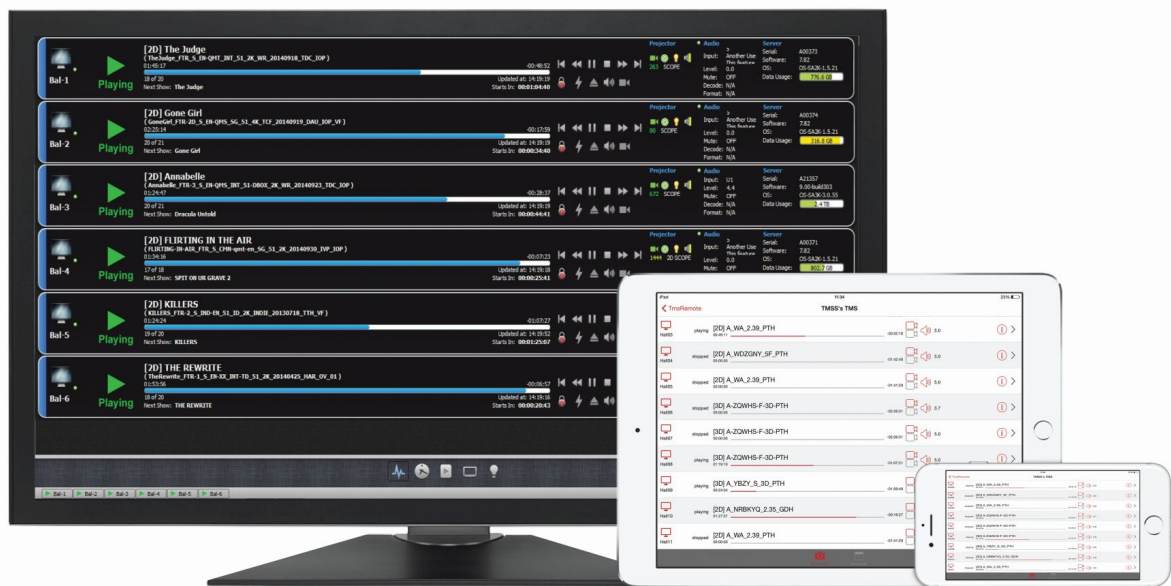
Firma CinemaNext nabízí i deriváty Melody TMS a to Symfonie a Orchestra, které pracují na stejném principu avšak s omezenými funkcionalitami. [18]

2.2.2 GDC TMS-2000

Jedná se o konkurenční systém společnosti *GDC*, která je zároveň i předním výrobcem kino-serverů a integrovaných media bloků. Hlavním rozdílem oproti Melody TMS je spolupráce pouze s vlastními zařízeními. Díky této koncepci je také výrazně nižší cena. Zákazník je ale zároveň nucen používat zařízení stejné firmy a při nutnosti výměny je nahradit nanejvýše jinými modely stejné značky.

Výhodou oproti Melody TMS krom nižší ceny je také fakt, že není třeba speciálního hardwaru v podobě serveru. TMS-2000 se instaluje na běžně dostupné platformy a operační systémy jako Linux nebo Windows.

Nevýhodou je, že neposkytuje okamžité řízení a vzdálené ovládání. Pouze programování show a správu obsahu. [19]



Obrázek 13: Ukázka systému GDC TMS-2000 na různých platformách, zdroj: gdc-tech.com

3 Bezdrátová komunikace a předávání dat

Bylo vytyčeno za cíl komunikovat s mobilním zařízením bezdrátově. Tím se rozumí některá z nabízených možností rádiového spojení, jehož adaptérem je běžně dostupné mobilní zařízení vybaveno. Cílem takové komunikace je výměna dat mezi dvěma zařízeními, v tomto případě mezi tabletem a serverem.

3.1 Bezdrátová komunikace s mobilním zařízením

Nejrozšířenějšími rozhraními, jejichž bezdrátovými adaptéry je většina mobilních zařízení vybavena, je Bluetooth, Wi-Fi, NFC a GSM. Technologie GSM se používá pro připojení k mobilní síti operátora a využití mobilního internetu. NFC pak pro výměnu dat a bezdrátové nabíjení na velmi krátké vzdálenosti, kde je nutný přímý dotyk obou zařízení. Navíc ne každé zařízení je technologií NFC vybaveno a některé levnější tablety nedisponují ani GSM rozhraním.

Technologie Bluetooth se v mobilních zařízeních začala vyskytovat daleko dříve než Wi-Fi. V době před nástupem chytrých mobilních telefonů se prostřednictvím Bluetooth přenášely kontakty, videa a obrázky. V dnešní době nalezne své uplatnění jako spojení pro příslušenství mobilních telefonů, jako jsou například chytré náramky, hodinky, trackery, bezdrátová sluchátka nebo klávesnice. Každé z těchto zařízení se používá v blízkosti mobilního telefonu. Dosah signálu Bluetooth je v uzavřených prostorech malý (do 20m) a pro použití ve větších budovách je nevhodný. [38]

Oproti tomu je technologie Wi-Fi vhodnější pro použití ve větších obestavěných prostorech. Prostřednictvím Wi-Fi se zařízení připojují přímo do lokální sítě, kde mají přístup k ostatním zařízením a také zařízením umístěných v internetu. To technologie Bluetooth přímo neumožňuje. V novostavbách se během projektování připravují a rozmisťují přístupové body (AP), které zlepšují pokrytí budovy bezdrátovým připojením k internetu. Protože aplikace bude kromě lokálního řídicího serveru komunikovat také s on-line databázovým serverem umístěným ve veřejné síti, jevílo se použití Wi-Fi jako vhodnější. [38]

3.2 Výměna dat prostřednictvím TCP/IP protokolu

Dle ISO/OSI modelu zařízení připojené do sítě prostřednictvím Wi-Fi navazuje spojení na úrovni transportní vrstvy pomocí TCP/IP protokolu. Zařízení se připojuje na server pomocí jeho IP adresy a portu, na kterém je spuštěna příslušná služba. Komunikace probíhá nešifrovaně a nezabezpečeně, čímž může být odposlechnuta a upravena.

Proto bylo rozhodnuto realizovat komunikaci se serverem na nejvyšší aplikační vrstvě pomocí zasílání HTTP, respektive HTTPS požadavků. Protokol HTTP rozlišuje 8 různých metod, jakými lze odeslat požadavek na server. Nejčastěji se používají metody GET a POST. První zmíněná umí pouze získat data ze serveru na základě parametrů URI. V praxi pak URL adresa může vypadat takto: `http://www.server.com/index.php?param1=text¶m2=1234`. Jedná se o pří-

klad předání dat prostřednictvím PHP skriptu. Za adresou serveru a názvem skriptu je uveden klíč prvního parametru a za znakem = pak jeho hodnota, ampersandem je oddělen další parametr. Výsledný HTTP požadavek pak vypadá takto.

```
GET /index.php?param1=text&param2=1234 HTTP/1.1
HOST: www.server.com
```

Výpis 1: HTTP požadavek zaslaný metodou GET

Metoda GET se pro předávání více parametrů moc nehodí a proto se data posílají metodou POST. Ta umožňuje poslat data přímo v HTTP požadavku (hlavičce) a není nutné je skládat do URI. URL adresa pak vypadá takto: `http://www.server.com/index.php`. Přičemž zařízení, které HTTP požadavek odesílá přidá data do jeho hlavičky. Výsledný HTTP požadavek pak vypadá takto.

```
POST /index.php HTTP/1.1
HOST: www.server.com
content-type: application/form
content-length: 22

param1=text
param2=1234
```

Výpis 2: HTTP požadavek zaslaný metodou POST

Oproti metodě GET přibyl identifikátor typu obsahu a také délka zasílaných dat, aby server mohl identifikovat, jestli již má k dispozici všechna data. Metodou POST se nemusí předávat jen parametry z formulářů. Obdobným způsobem se nahrávají na server soubory nebo se předávají řetězce ve formátu JSON. [20]

3.3 Serializační formát JSON

Serializační formát JSON vznikl a je určen pro výměnu dat ve formě objektů mezi zařízeními. JSON je jednoduše čitelný pro uživatele i pro strojové čtení. Ačkoliv je založen na programovacím jazyku JavaScript, je jeho použití umožněno nezávisle na jiném programovacím jazyku nebo platformě.

JSON je založen na dvou strukturách. Jednou z nich je kolekce párů název/hodnota. Každá hodnota má svůj klíč, který není jen číselným indexem, jako je tomu v případě klasického pole hodnot. Název klíče může přímo korespondovat s názvem sloupce v databázi nebo konkrétní vlastností. Taková struktura, která se nazývá JSON objekt je v pokročilých programovacích jazycích lehce převeditelná na slovníky, mapy, asociativní pole a další generické třídy. Druhou používanou strukturou je seřazený seznam hodnot. Řazení není prováděno pomocí klíče (ten chybí), ale právě podle určitých vlastností hodnot obsažených ve struktuře. Například podle

abecedy, podle číselné hodnoty atd. Tato struktura se nazývá JSON pole a lze ji realizovat pomocí polí, vektorů, listů, seznamů nebo posloupností. [21]

3.3.1 Datové typy JSON

Kromě výše zmíněných struktur umožňuje formát JSON použití dalších datových typů: [21]

- **Číslo** je číselně vyjádřená hodnota v dekadickém tvaru.
- **Řetězec** je několik tisknutelných znaků následujících po sobě uzavřených uvozovkami.
- **Binární hodnota** může nabývat hodnot *true* nebo *false*.
- **Nulový datový typ** známý také jako *null*.
- **JSON objekt** obsahuje několik párů kolekce klíč/hodnota. Objekt začíná a končí složenými závorkami. Klíč je vždy řetězec uzavřený uvozovkami, po klíči následuje dvojtečka a po ní hodnota. Ta může nabývat jakýchkoliv datových typů podporovaných formátem JSON, včetně dalšího JSON objektu. Pořadí párů ve formátu JSON není rozhodující.
- **JSON pole** obsahuje několik hodnot stejného datového typu. Prvkem pole může být i další JSON pole. Pole začíná a končí hranatými závorkami, jednotlivé prvky se oddělují čárkami a jsou adresovatelné pomocí číselného indexu začínajícího od nuly.

```
{  
  "souřadnice": [  
    1.54,  
    4.63,  
    4.45,  
    9.45  
  ],  
  "status": true,  
  "barva": "#82b92c",  
  "null": null,  
  "číslo": 123,  
  "objekty": {  
    "a": "b",  
    "c": "d",  
    "e": "f"  
  },  
  "řetězec": "Hello World"  
}
```

Výpis 3: Příklad JSON objektu

4 Problematika značení DCP balíků

Jak již bylo zmíněno výše, norma DCI řeší komplexně veškeré technické záležitosti spojené s digitální projekcí. Mezi její části patří také normalizovaný tvar názvu DCP balíků. Norma v tomto ohledu není jako ostatní normy závazná, ale doporučující. I když soubor není pojmenován dle normovaného názvu, server je schopen jej přecíst a správně přehrát. Pro obsluhu i server ale nemusí být na první pohled patrné vlastnosti takového souboru.

4.1 Struktura DCP balíků

DCP balík je adresář, který se skládá ze souborů. Mezi soubory, které jsou povinné patří tyto: [22]

- **CPL** (Composition Playlist) je XML soubor, který je nositelem všech atributů o DCP. Obsahuje jeho název, typ, délku, datum vytvoření, datum platnosti. Rovněž určuje v jakém pořadí se budou přehrávat obrazové a zvukové soubory. Jedná se o hlavní soubor celého balíku DCP.
- **PKL** (Packing List) jedná se opět o XML soubor obsahující hash, ze kterého kino-server pozná správné pořadí přehrávaných audio a video souborů.
- **Obrazové a audio soubory** jsou reprezentovány několika MXF soubory, které umožňují i použití metadat. Jelikož jsou obrazové i zvukové soubory pro zachování nejvyšší kvality uchovávány bez komprese a v nativním rozlišení, musí být jednotlivé AV části umístěné do více souborů (reelů). Operačnímu systému se rovněž lépe pracuje s více menšími soubory než s jedním velkým. Oproti předchozím typům souborů, kterých bývá v DCP po jednom.

Pokud je DCP s titulky, může obsahovat textový či jiný běžně používaný titulkovací soubor s metadaty.

4.2 Systematika značení DCP balíků a jejich atributů

Stejně jako každý soubor má i DCP své vlastnosti (atributy). Dle normy se rozlišuje několik vlastností, které jsou stěžejní pro správné nasazení a odbavení titulů. Aby byly vlastnosti identifikovatelné ihned ze samotného názvu souboru, pojmenovává se DCP textovým řetězcem, který obsahuje podtržítka oddělené vlastnosti, které reprezentují dohodnuté zkratky v daném pořadí. Některé z nich jsou volitelné, některé naopak povinné. [22]

Název titulu: Maximální povolená délka je 14 znaků. Nepoužívá se označení 3D v názvu, pokud je titul ve 3D. Jednotlivá slova se oddělují velkými písmeny. Pro značení pořadí titulů se číslice neoddělují pomíčkou ani podtržítkem. Neměly by se používat zkratky.	Poměr stran obrazu: F = Flat S = Scope C = Plný kontejner Pokud má formát odlišný poměr stran, uveďte se tento poměr za pomíčkou	Jazyk: První dva nebo tři znaky indikují jazyk zvuku, další dva nebo tři znaky po pomíčce indikují jazyk titulků. Jsou-li titulky značeny malými písmeny, pak jsou titulky vypáleny v obraze, pokud jsou značeny velkými písmeny, jsou vkládány projektořem. Značení OCAP nebo CCAP po pomíčce indikuje uzavření některých částí překladu.	Rozlišení obrazu: 2K nebo 4K	Název softwaru: Software, který DCP vyrobil	Verze: OV = originální verze VF = verifikovaná verze
		Studio: Dva až čtyři znaky pro název studia, které snímek produkovalo	Datum výroby: Datum výroby ve formátu YYYYMMDD	Norma: IOP = Interop SMPTE = SMPTE po pomíčce 3D	

NazevTitulu_TLR-1-Temp-RedBand-Chain-3D-4-fl-48-DVis_F-133_EN-EN-OCAP_US-GB_51-HI-VI-IAB-Dbox_2K_ST_20190103_Facility_IOP-3D_OV

Typ obsahu: FTR = film EPS = epizoda TLR = trailer TSR = teaser PRO = promo RTG = hodnocení / omezení RTG-F = omezení pro film RTG-T = omezení pro trailer POL = upozornění PSA = oznámení veřejné služby ADV = reklama SHR = krátký film XSN = přechody TST = testy po pomíčce mohou následovat verze obsahu	Modifikátory typu obsahu: Temp = předběžná verze bez finálního obrazu nebo zvuku Pre = Před-premiérová verze s finálním obrazem a zvukem RedBand = omezení pouze pro trailery s obsahem nevhodným pro děti a mládež Chain = Událost, pro kterou bylo DCP speciálně vyrobeno. 3D = 3D 2D = 2D verze obsahu, který také existuje ve 3D 4fl = svítivost, při které je zajištěno správné barevné podání 48 = 48 FPS, poze pokud je jiné než 24 DVis = Dolby Vision gradace obrazu EC = Eclaircolor	Území a přístupnost: První dva nebo tři znaky indikují zemi dle kódu ISO 3166-1, pro kterou je DCP určeno, po pomíčce následuje omezení přístupnosti	Formát zvuku: 51 = 5.1 71 = 7.1 10 = mono 20 = stereo (bez Lt a Rt) HI = audio stopa pro neslyšící VI = audio popis pro nevidomé SL = znakový jazyk IAB = Immersive Audio Bitstream (ST2098-2) Atmos = Dolby Atmos Auro = Barco Auro DTS-X = DTS-X Dbox = pohybový simulátor
---	---	--	---

Obrázek 14: Popis názvu DCP balíku, zdroj: autor

- Pro **Název titulu** je dovoleno použít maximálně 14 znaků ze základní ASCII tabulky. Pokud je název delší, použije se známá zkratka titulu. Pro víceslovné názvy se používají pomlčky nebo velká písmena na začátku každého dalšího slova (camell case).
- **Typ obsahu** rozlišuje o jaký typ audiovizuálního díla se jedná. Rozlišuje se celkem 12 typů:
 - FTR (Feature) - celovečerní film.
 - TLR (Trailer) - upoutávka k celovečernímu nebo krátkému film.
 - TRS (Feature) - upoutávka ve formě hrubého sestřihu, většinou z natáčení, bez kreditů.
 - EPS (Episode) - epizoda seriálu.
 - PRO (Promo) - reklamy na nabízené služby.
 - RTG (Rating) - upozornění na omezení přístupnosti před filmem.
 - POL (Policy) - slouží pro kodex chování. Upozorňuje na vypnutí mobilních telefonů, zakazuje pořizování záznamů atd.
 - PSA (Public Service Announcement) - ohlášení veřejné služby. Upozorňuje diváky o veřejných sbírkách, konání voleb atd.
 - ADV (Advertisement) - reklama.
 - SHR (Short) - krátký film, předfilm (do cca 20 minut).
 - XNS (Transitional) - velmi krátké většinou černé klipy pro přepnutí a vyplnění místa mezi jednotlivými soubory.
 - TST (Test) - testovací obrazce pro nastavení rámování, ostření, zkoušku zvuku atd.

Typ obsahu může být také doplněn o sdělení, zda se jedná o 2D nebo 3D verzi, předběžné, režisérské nebo finální vydání, jestli je použit vyšší frame rate nebo vysoký dynamický kontrast.

- **Obrazový formát** Určuje o jaký formát se jedná.
 - F (Flat) - 1998 x 1080 px, respektive 3996 x 2160 px v případě 4K. Může být doplněno o číslo, které udává modifikovaný formát založený na výšce obrazu. Například F-133 udává poměr stran, kdy výška obrazu v pixelech je 1080, respektive 2160 a šířka je 1,33 krát větší než výška (cca 1436 px, respektive 2872 px v případě 4K rozlišení) Používá se pro digitalizaci starších a atypických formátů. Standardní poměr stran u Flatu je 1:1,85
 - S (Scope) - 2048 x 858 px, respektive 4096 x 1716 px v případě 4K. Může být rovněž doplněno o číslo, které udává modifikovaný formát založený na výšce obrazu. Například S-235 udává poměr stran, kdy šířka obrazu v pixelech je 2048, respektive 4096 a výška je 2.35 krát menší než výška (cca 871 px, respektive 1742 px v případě 4K rozlišení). Standardní poměr stran Scope je 1:2,39
 - C (Full container) - 2048 x 1080 px, respektive 4096 x 2160 px pro 4K.
- **Jazyk a lokalizace** určuje jaký je použitý jazyk zvuků, titulků nebo pro který stát je dané DCP určeno. Používá se dvoumístná zkratka korespondující se zkratkou pro mezinárodní identifikaci dle ISO 3166-1. Existují také zkratky jako CS, která značí společnou lokalizaci pro Českou i Slovenskou republiku nebo QBP pro mexickou španělštinu. V případě, že je zkratka uvedena malými písmeny, jsou titulky vypáleny v obraze a nejde s nimi dále manipulovat. V ostatních případech vkládá titulky projektor ze zvláštního souboru. Kompletní seznam zkratek je dostupný na stránkách <http://isdcf.com/dcnc/>.
- **Zvukový formát a prostorové efekty** se značí dvojcifernou číslicí, kdy první znamená počet plno rozsahových kanálů a druhé počet speciálních efektových kanálů. 51 tak znamená známou konfiguraci 5.1 (levý, pravý, střed, levý surround, pravý surround, efektový subwoofer), 20 pak 2.0 (stereo). Minimální konfigurace pro DCI kino je 5.1. V případě 7.1 a vyšších prostorových zvuků je třeba vzít v potaz, že zadní surroundové kanály nebudou reprodukovatelné. Další zkratky jako Atmos, Auro nebo DTSX označují licencované prostorově orientované objektové zvukové formáty. DBOX pak efek vibrujících sedadel a další prvků sálu.
- **Rozlišení obrazu** dle normy DCI v současnosti připouští pouze 2 druhy rozlišení, a to nativní 2K a vyšší 4K. Pro odbavování obsahu s vyšším rozlišením než 4K, nejsou dostupné technologie ani dosud specifikované standardy.
- **Produkční studio**, které daný titul vyrobilo a je primárním držitelem autorských práv na audiovizuální dílo. Používají se zkratky o dvou až čtyřech znacích.

- **Datum výroby** značené pomocí 8-mi místného čísla, kde první čtyři místa reprezentují rok a zbylá místa po dvou měsíc a den včetně vodících nul.
- **Post produkční studio**, které se podílelo na postprodukci. Je výrobcem DCP, odpovídá za jeho kompatibilitu, hlídá a řídí vystavování KDM.
- **Standard**, podle kterého je DCP vyrobeno. Starším a plně podporovaným je IOP a novější, na kterou se snaží filmový průmysl kvůli plné kompatibilitě s ostatními distribučními platformami (televize, videotéky atd) přejít - SMPTE.
- **Verze DCP** rozlišuje mezi OV (originální verzí) a VF (pozměněnou verzí), případně jejími dalšími verzemi. Originální verze je nositelem obrazu a zvuku. Pozměnění verze je v porovnání s originální verzí podstatně menší a obsahuje navíc třeba jen titulky a 2 zvukové kanály pro rozšíření z 5.1 na 7.1. Tímto dělením se šetří na disku místo a usnadňuje se manipulace. Pokud kino divákovi nabízí titul v původním znění, s titulky, s dabingem nebo v další dostupné variantě, nemusí stejný titul uchovávat na disku vícekrát. Postačí mu jeden hlavní a další doplňující. Z popisu vyplývá, že VF verze není sama o sobě přehratelná.

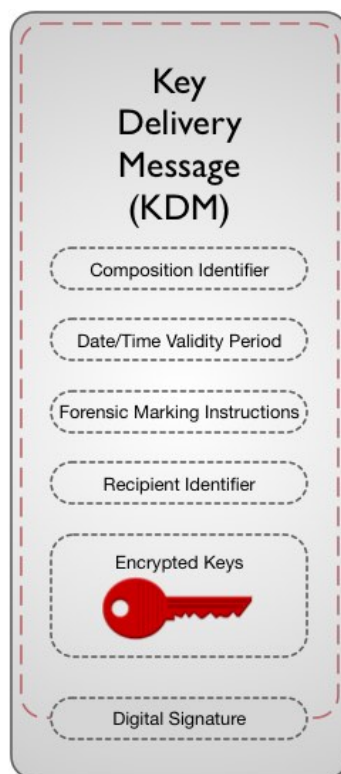
Promítači je po přečtení názvu DCP jasné, o jaký typ se jedná, kam jej v playlistu zařadit, jak nastavit reprodukci zvuku a obrazu. Některé servery disponují funkcí pro automatickou identifikaci DCP a na základě jeho názvu nastaví projekci automaticky. Pokud je ale z výroby DCP špatně pojmenováno, je následně i divákovi špatně odbaveno, není-li provedena zkušební projekce. Proto tuto funkci nalezneme v kinech většinou vypnutou a není ani příliš vyhledávaná.

Zejména při výrobě upoutávek se nehledí na obrazový formát a vše je označeno jako FLAT, protože pokud na tento formát odbaví upoutávka ve SCOPE, ztratí se nanejvýše několik sloupců pixelů z levé a pravé strany, s jejichž ořezem se počítá už při natáčení. V opačném případě (flatová upoutávka na SCOPE) se získá pouze nezajímavý výřez.

4.3 Verzování a omezování platnosti DCP souborů

Samotný DCP balík má krom data pořízení také datum expirace. Neznamena to, že by po jeho vypršení byl balík nehratelný. Jedná se o predikci a předpoklad, do kdy bude možno tento balík vzhledem k vývoji technologií a úprav standardů korektně přehrát. [37]

Je-li třeba omezit platnost DCP, používá se již dříve zmíněné KDM. Mezi jeho vlastnosti patří doba platnosti a sériové číslo serveru, pro které je určen. Název není nijak standardizován, měl by ale vystihovat titul, pro který je určen. KDM je pouze nositelem symetrických zašifrovaných klíčů. Samotné KDM je nezašifrované a navíc se s ním nemusí zacházet zvláště opatrně. Právě naopak, může se s ním zacházet naprosto otevřeně. Lze jej vystavit na webové stránky, zaslat do e-mailu atd. Pokud se dostane k nesprávnému příjemci, bezpečnost není ohrožena. KDM není určeno pro jeho cílové zařízení a s DCP se nespojí. Výhodou jeho "otevřenosti" je i jednoduchá diagnostika nespolečujícího klíče. V textovém editoru se soubor klíče otevře a zjistí se jeho nefunkčnost. Jiné DCP, jiné sériové číslo kino-serveru atd.



Obrázek 15: Systém KDM ochrany, zdroj: cinepedia.com

Důvěra ochrany je plně vložena do výrobců kino-serverů a projektorů. Pokud dojde k porušení šasi projektoru, kino-serveru nebo jiné části řetězce, okamžitě se zablokují pro promítače veškeré DCP chráněné KDM. V tento okamžik nezáleží ani na samotné platnosti KDM a jeho určení. Dokud servisní technik nebo jím pověřená osoba neprovede kontrolu zařízení proti nechtěnému nahrávání obsahu jiným zařízením, či jiné porušení nebo nepovolenou úpravu technologie, která by mohla kráť práva držitele autorských práv, je samotným serverem obsah uzamknut. Servisní technik musí toto potvrdit zadáním servisního kódu či přiložením servisního čipu bezprostředně na hardware. Nelze provést vzdáleně.

Každý media-blok je vybaven dvojicí baterií pro velmi přesný obvod RTC s nepřepisovatelným nastavením. Při výrobě je do něj nahrána hodnota aktuálního času. Integrátor může provádět korekci v maximálním rozsahu ± 6 minut. Pokud dojde k provozu při úrovni napětí na bateriích pod dovolenou mez, začne se RTC výrazněji rozcházet. Vystavené KDM pak nemusí časově sedět s reálnou dobou projekce. Pokud nedojde k jejich včasné výměně, stává se media-blok již neopravitelným. [37]

Jediným ohrožujícím faktorem tak zůstává samotné vystavování klíčů a jejich evidence. To se většinou děje z jednoho centrálního místa. Pro zahraniční tituly je nejznámějším distributorem KDM společnost *Deluxe Technicolor*.

5 Databázový systém

Moderním nástrojem pro efektivní správu a nakládání s daty jsou relační databázové systémy. Data jsou v databázi uchovávána v tabulkách, které sdružují jednotlivé záznamy. Každá tabulka má více sloupců. Každému sloupci náleží stejná vlastnost společná pro všechny záznamy (např. barva, věk, rozměry atd.). Některé sloupce chápeme jako cizí klíče, které odkazují na jiné sloupce v jiné tabulce ve stejné databázi. Vztah klíčů mezi sebou se nazývá relací. Díky relaci můžeme vytvářet různé sestavy, které čerpají data z vícero tabulek. [32]

5.1 Výhody a nevýhody jednotlivých SQL systémů

V současné době existuje několik hotových řešení databázových systémů, které nabízejí využití pro široké spektrum aplikací. Díky velkému výběru databází není efektivní vyvíjet vlastní systém pro ukládání dat. Je vhodnější použít hotové a odlazené řešení, za kterým stojí detailní dokumentace, komunita uživatelů a podpora výrobce.

S databází se komunikuje pomocí SQL příkazů. Syntaxe SQL byla navržena tak, aby byla intuitivní a přívětivá pro uživatele. Pomocí SQL příkazů lze vkládat, upravovat, číst nebo mazat data. Rovněž lze vytvářet tabulky a databáze, přiřazovat tabulkám vlastnosti, klíče a databázím tabulky či uživatele.

V následující tabulce uvádím přehled nejznámějších databázových systémů, včetně srovnání jejich vlastností.

Tabulka 1: Srovnání základních vlastností relačních databázových systémů

Databázový systém	Licence	Současný vlastník	Dostupné platformy	Počet dotazů za sekundu
Microsoft SQL Server	EULA	Microsoft	Windows	3785,76
MySQL	GNU	Oracle Corporation (od 2008)	Windows, Linux	1701,28
PostgreSQL	MIT	PostgreSQL Global Development Group	Mac OS X, Windows, Linux, Solaris	3688,68
Oracle	OLMS	Oracle Corporation	multiplatformní	885,94
Firebird	GNU	Firebird Foundation	Windows, Linux, MacOSX, Solaris	2087,8

Z rešerše vidíme, že nejvýkonnějším ze srovnávaných databázových systémů je Microsoft SQL Server. Pro zamýšlený účel není výkon naprostou prioritou. Databáze bude obsahovat minimum tabulek a relací. Jejich obsluhu by tak měl s uspokojivým výkonem zvládnout i některý z volně dostupných systémů. Firebird má oproti ostatním výhodu, že jej lze přímo nainstalovat na zařízení běžící pod operačním systémem Android. Komunikace s databází bude u tohoto systému pak daleko jednodušší než u ostatních, neboť nebudeme muset využívat aplikačního

serveru. Všechny databáze krom první zmiňované jsou šířeny pod různými obdobami volných licencí, jejich použití je tedy pro tento účel bezplatné.

V požadavcích na aplikaci bylo vytyčeno použití databázového serveru dostupného i pro případné aplikace na jiných platformách. To by s databází, která je spuštěna na jednom zařízení Android bylo obtížnější.

Srovnání výkonu vychází z analýzy. [27] Ke správné interpretaci výsledků je zejména nutné přihlídnout ke konkrétnímu použitému hardwaru, jeho nastavení a optimalizaci databázového systému.

5.2 Databázový systém MySQL

Po provedené rešerši bylo rozhodnuto pro použití databázového systému MySQL. Jedná se o nejčastější systém, který lze v nabídce komerčních webových hostingů objednat a provozovat bez větších omezení. To je možné především díky jeho volně šiřitelné licenci GNU. Navíc má tato databáze širokou podporu různých skriptovacích frameworků a v neposlední řadě také kvalitní dokumentaci. Databáze MySQL lze nainstalovat pro vývojové potřeby jako součást balíků XAMPP na kterýkoliv počítač se systémem Windows. Balík obsahuje webový server Apache2, databázi MariaDB, která sdílí jádro s MySQL a engine pro vykonávání skriptů psaných v jazyce PHP.

Databáze MySQL vznikla v roce 1985, kdy programátoři Michael Widenius, David Axmar a Allan Larsson založili švédskou firmu *MySQL AB*. V roce 2008 pak firmu společně s databází koupila společnost *Sun Microsystems*, která spadá pod *Oracle Corporation*.

MySQL nabízí velkou variabilitu datových úložišť. Oproti konkurenčním systémům lze kombinovat různá úložiště jak samotných databází, tak i jednotlivých tabulek. Uživatel tak může pro každou tabulku vybrat jiné úložiště a optimalizovat tím výkon celého systému. Mezi nejčastěji používané typy úložiště patří: [27]

- MyISAM
- InnoDB
- Memory (heap)
- Falcon
- solidDB

5.2.1 Datové úložiště MyISAM

MyISAM je jedno z nejstarších datových úložišť v MySQL, které vychází z jeho předchůdce ISAM. Podporuje fulltextové vyhledávání, ukládání a indexaci prostorových dat nebo indexaci prvních 500 znaků u záznamů typu text a blob. Maximální velikost úložiště je 256 TB.

Technologie úložiště MyISAM uchovává každou tabulku jako dva samostatné soubory: [27]

- *.myd – soubor obsahující data tabulky.
- *.myi – soubor obsahující indexy tabulky.

Definice tabulky je poté uložena v souboru *.frm, který náleží přímo MySQL serveru. [27]

Od úložiště MyISAM se přechází k novějšímu úložišti InnoDB, které na rozdíl od výše zmíněného podporuje cizí klíče a transakční zpracování dotazů. Tato funkcionality snižuje rychlost databáze, ale zato zajišťuje konzistenci dat. Pokud není kladen důraz na konzistenci dat v tabulkách, může se jevit použití staršího MyISAM jako vhodnější.

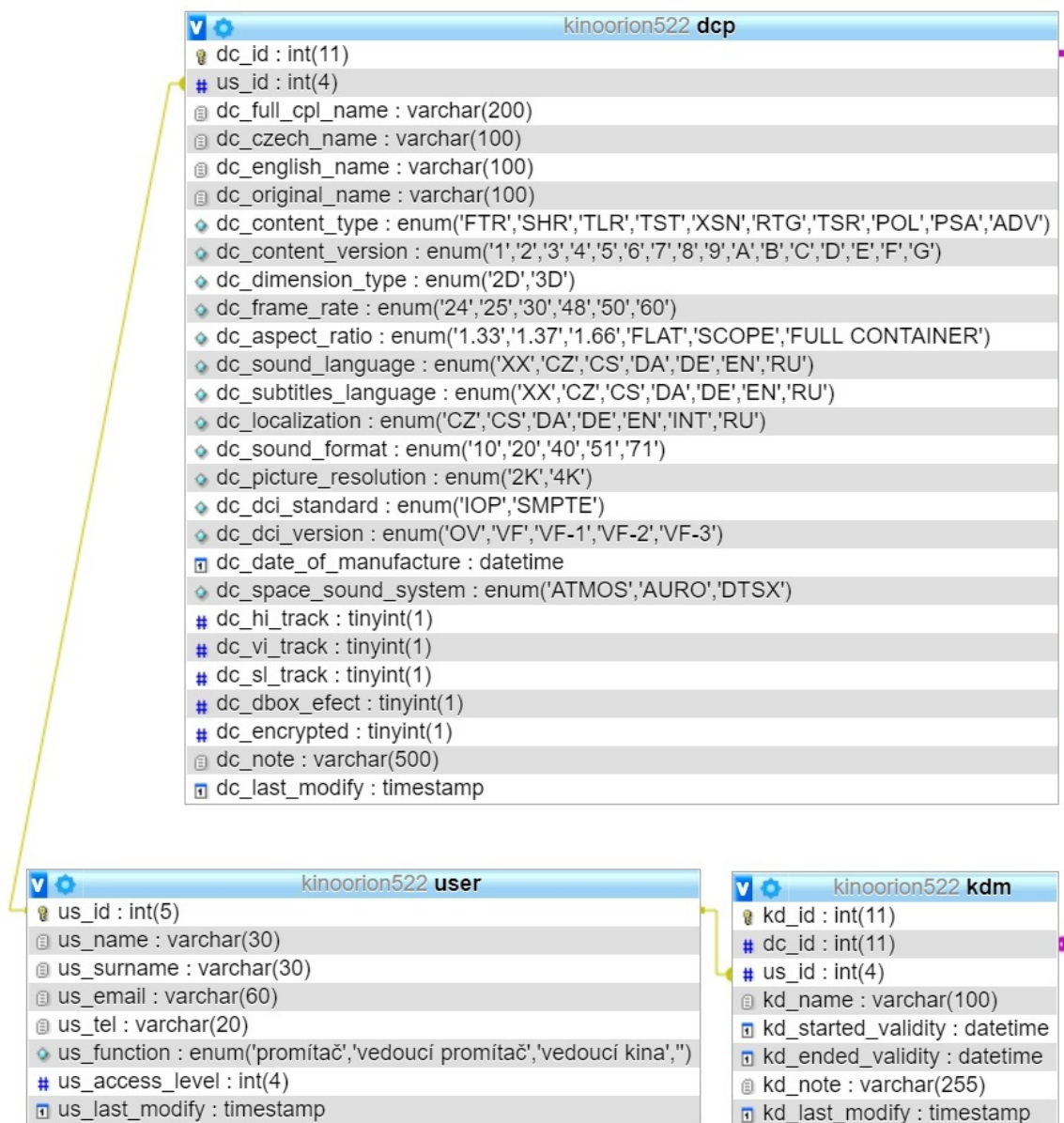
5.2.2 Datové úložiště InnoDB

Toto úložiště bylo navrženo pro MySQL firmou *Innobase Oy*, odkud je odvozen i jeho název. Oproti výhodám zmíněných v předchozí kapitole, toto úložiště dále umožňuje zamykání jednotlivých sloupců tabulky. Neumožňuje ale indexaci prostorových dat a chybí také fulltextové vyhledávání.

5.3 Návrh struktury databáze

Úložiště pro databázi bylo zvoleno typu InnoDB, jehož vlastnosti a výhody byly popsány v předchozí kapitole, protože databáze pracuje s vazbami a rovněž používá transakce, díky nimž je třeba zajistit atomicitu, konzistenci, izolovanost a trvalost. Souhrnně se tyto vlastnosti označují zkratkou ACID. [28]

Po provedené analýze byl vytvořen návrh struktury databáze. Tu tvoří celkem tři tabulky. V tabulce dcp jsou uloženy všechny vlastnosti DCP balíků. Ke každému DCP, které je zabezpečeno pomocí KDM, lze přiřadit nekonečně mnoho klíčů (KDM) z tabulky kdm. Proto bylo použito relace 1:n. Každému KDM a DCP lze rovněž z tabulky user přiřadit uživatele, ve smyslu autora záznamu nebo jeho poslední úpravy. Proto i zde je relace 1:n, kdy každý uživatel může být autorem nekonečně mnoho DCP a KDM. Struktura databáze je zobrazena na diagramu níže:



Obrázek 16: E-R diagram databáze a datový slovník, zdroj: autor

Název tabulek a sloupců byl volen v jednotném čísle. Všechny názvy jsou psány malými písmeny, slova se oddělují podtržítkem. Každý sloupec začíná prefixem, který určuje jaké tabulce náleží. Prefix *dc_* náleží tabulce *dcp*, *kd_* tabulce *kdm* a *us_* tabulce *user*. Pokud se v tabulce vyskytuje sloupec s jiným prefixem než náleží dané tabulce, jedná se o cizí klíč s referencí na jinou tabulku.

5.3.1 Tabulka *dcp*

Z E-R diagramu je patrné, že tabulka *dcp* obsahuje celkem 27 sloupců z toho jeden sloupec je index cizího klíče odkazující na index autora (*us_id*) v tabulce *user*. Sloupec *dc_id* má nastavený příznak automatické inkrementace. Při vkládání nového záznamu tak není nutné zjišťovat poslední hodnotu indexu a ani ji zadávat. Databáze ji doplní sama. Sloupce pro zadání názvů a poznámky jsou typu *varchar*. Sloupec *dc_czech_name* má oproti ostatním nastavené porovnávání *utf8_czech_ci*, které při transakci k seřazení záznamů dle tohoto sloupce akceptuje českou abecedu. U vlastností, kde je jejich výčet předem znám je použit datový typ *enum*. Ty lze jednoduše modifikovat skrze název nebo jako referenci. Dalším datovým typem je *tinyint*, který uchovává binární hodnotu. Sloupec data výroby *dc_date_of_manufacture* je datového typu *datetime*, sloupec *dc_last_modify* - datum poslední úpravy pak *timestamp*, přičemž jeho hodnota je aktualizována automaticky při změně kterékoliv hodnoty.

5.3.2 Tabulka *kdm*

Tabulka *kdm* má podstatně méně sloupců (8) než tabulka *dcp*. Sloupec *kd_id* má také nastavený příznak automatické inkrementace. Sloupce *dc_id* a *us_id* jsou cizí klíče, které odkazují na DCP, ke kterému je KDM přiřazeno a také na uživatele, který je autorem poslední změny. Sloupce *kd_started_validity* a *kd_ended_validity* jsou datového typu *datetime*. *kd_last_modify* datového typu *timestamp*, přičemž jeho hodnota je aktualizována opět zcela automaticky při jakékoliv úpravě.

5.3.3 Tabulka *user*

Třetí tabulkou je tabulka uživatelů, která obsahuje 8 sloupců a nese údaje o jednotlivých uživateliích systémů. Sloupec *us_id* má opět nastavený příznak automatické inkrementace. Nachází se zde také sloupec *us_function* datového typu *enum*. Ten určuje pracovní pozici uživatele v rámci kina. Každému uživateli lze také přiřadit úroveň pomocí hodnoty ve sloupci *us_access_level* datového typu *int*, na základě které lze uživateli přiřadit patřičná oprávnění.

5.4 Komunikace s databází MySQL

S databází MySQL stejně jako s ostatními systémy je možné komunikovat několika různými způsoby. Buď pomocí klasické konzole, pomocí protokolu TCP/IP nebo skrze aplikační server.

5.4.1 Komunikace a ovládání pomocí konzole

Mezi ten nejzákladnější patří komunikace pomocí konzole (terminálu) ať už vzdáleně nebo přímo na počítači, na kterém databázový server běží. Spíše než pro samotnou komunikaci se používá zejména pro instalaci, správu a údržbu databáze. Samozřejmě lze zadávat i SQL příkazy pro manipulaci s daty, ale vzhledem k napojení na ostatní aplikace není moc využíván. [32]

5.4.2 Komunikace pomocí protokolu TCP/IP

Nejvyužívanější možností komunikace se serverem je přes síť. Databázový server naslouchá ve výchozím nastavení na portu 3306. Síťová komunikace představuje univerzální komunikační adaptér, na který se mohou napojit veškerá zařízení, která dokáží pomocí síťového TCP/IP protokolu komunikovat. V praxi se jedná o počítače, IoT, mobilní zařízení nebo třeba embedded systémy s mikroprocesory. Každé zařízení běžící na odlišné architektuře potřebuje pro napojení na server knihovnu, která překládá komunikaci na serveru známé požadavky. Nevýhodou této komunikace je nebezpečí tzv. SQL injection. Ne vždy musí být známá celá struktura sítě, přes kterou bude zařízení a server navazovat spojení. Nemůžeme tak nikdy vyloučit, že požadavek nebude během cesty na server upraven a v databázi nevznikne šum, případně dojde ke ztrátě dat. Samozřejmě existuje možnost zabezpečeného SSL spojení. Jeho implementace na klienta, zvláště jedná-li se o zařízení s elementární síťovou funkcí, nemusí být vždy jednoduchá. [32]

5.4.3 Komunikace pomocí aplikačního serveru

Pro potřeby tohoto projektu bylo rozhodnuto o komunikaci pomocí aplikačního serveru. V praxi to funguje tak, že využije webového serveru, který vytvoří prostředníka mezi klientem a databázovým serverem. Klient odesílá požadavky formou HTTP případně zabezpečeného HTTPS protokolu na webový server. Ten spustí požadovaný skript se vstupními daty, který vyvolal klient. Skript může provést předem naprogramovanou úpravu dat a posléze samotnou operaci s daty v databázi. Klientovi poté vrátí odpověď. Výhodou použití aplikačního serveru je zabezpečení. Databázový a webový server běží častokrát na jednom hardwaru nebo v interní zabezpečené síti, čímž je vyloučeno napadení komunikace přímo před databází. Aplikační server také může zpracovávat pouze požadavky, které mu naprogramujeme. Rozhodneme-li se neimplementovat příkazy pro mazání dat, je vyloučeno, aby jej klient nebo útočník z databáze smazal. Toto řešení je velkým usnadněním především pro zařízení, která mají omezenou funkcionalitu, protože o všechny složitější úpravy dat a jejich převod na SQL příkazy se stará skript běžící na výkonnějším hardwaru.

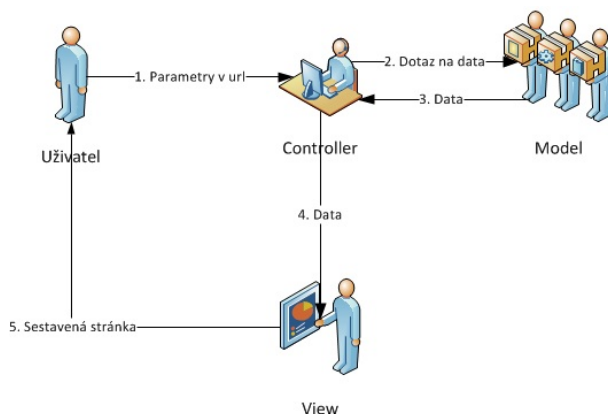
V příloze D.1 se nachází SQL soubor pro vytvoření struktury databáze a naplnění jí vzorovými daty. Osobní údaje a přístupová hesla v tabulce *user* jsou smyšlená.

6 Návrh a implementace aplikačního serveru

Pro provoz databázového a aplikačního webového serveru byl vybrán komerční webový hosting. Jeho součástí je jedna MySQL databáze a neomezený prostor pro nahraní zdrojového kódu aplikačního serveru. Pokud se na server odkážeme přímo pomocí webového prohlížeče, server rozpozná, že se nejedná o aplikaci (neobsahuje JSON požadavek), která má patřičná povolení pro komunikaci a přesměruje uživatele.

Aplikační server je postaven na frameworku Nette, který stojí na skriptovacím jazyce PHP. Použitá verze frameworku je 2.2.4. Framework již obsahuje předem připravené metody pro komunikaci s databází, parsování sérializačních formátů a také vlastní šablonovací systém Latte, sloužící pro renderování obsahu stránky (v našem případě odpovědi severu).

Framework Nette je určen ke tvorbě aplikací postavených na platformě MVC (Model-View-Controler), který respektuje pravidla pro oddělení jednotlivých částí kódu aplikace od sebe a umožňuje starší procedurální programování nahradit modernějším, objektovým. Princip zpracování požadavků aplikace postavené na modelu MVC je patrný z následujícího obrázku.



Obrázek 17: Data-flow diagram MVC aplikace, zdroj: itnetwork.cz

6.1 Směrování a routování

Příchozí požadavek je odbaven tzv. routerem. Je to část programu spadající do controlleru, která odbaví parametry zadané v URL a předá řízení aplikaci presenteru, který již obsahuje vlastní kód.

```
public static function createRouter(){
    $router = new RouteList;

    $router[] = new Route('<presenter>/<action>', 'Main:receive');

    return $router;
}
```

Výpis 4: Konfigurace routování URL v souboru RouterFactory.php

Je vytvořen `RouteList` a do něj přidána jedna routa. Ve složitější aplikaci jich může být více, v našem případě postačí jedna. Zápis v argumentu třídy `Route` znamená vzorový překlad relativní části adresy. Tedy místo kde obvykle stojí `index.php` nebo `kontakt.html`. Nette využívá tzv. pretty URL, tedy hezkých URL, které neprozrazují návštěvníkovi celou adresu, ale pouze jakýsi relativní odkaz. V námi popsaném případě je nutné uvést presenter a konkrétní akci, která se má vykonat. V případě, že je neuvedeme, proběhne presenter a akce zadaná jako výchozí. Pokud tedy chceme vyvolat činnost presenteru a dostat se tak blíže k databázi, použijeme adresu `https://app.server.com/main/receive`, respektive jen `https://app.server.com`.

6.2 Zpracování požadavku

Samotné zpracování požadavku probíhá v presenterech. Tato aplikace i díky své jednoduchosti využívá pouze jeden presenter. Routování je nastaveno tak, že se vždy volá metoda `actionReceive`. Zde je nejprve získán celý požadavek od klienta, ze kterého je následně vytaženo jen tělo, které pravděpodobně obsahuje příkaz ve formátu JSON. Objekt JSON je poté dekodován na asociativní pole, se kterým se dále pracuje. Nedílnou součástí každého příkazu je položka `command`, která obsahuje informaci o tom, jaký příkaz je třeba vykonat. Ve struktuře switch je poté příkaz jednoznačně identifikován a vyvolána metoda daného modelu. Metoda následně vrátí data opět v podobě asociativního pole, které je následně převedeno zpět do formátu JSON. Ten je již jako řetězec vytištěn do šablony, která je společně s odpovědí serveru vyrenderována. V případě, že během procesu vznikne chyba, vrací server příznak chybového zpracování požadavku, avšak v podobě JSON odpovědi s HTTP kódem 200. Klient pak na základě toho identifikuje, že se jedná o chybu na úrovni zpracování požadavku (např. chybný příkaz) a ne jedná se o chybu serveru (kód 404 nebo 500). Na výše popsaném postup lze ukázkově vidět, že presenter je jakýmsi zprostředkovatelem a samotná data a jejich interpretace se odehrávají v modelu.

```
public function actionReceive() {
    $httpRequest = $this->getHttpRequest();
    try {
        $arrayRequest = Json::decode($httpRequest->getRawBody(), Json::FORCE_ARRAY);

        switch($arrayRequest['command']){
            case 'insert_dcp':          $arrayResponse = $this->clipManager->insertDcp(
                $arrayRequest);
                break;
            case 'update_dcp':         $arrayResponse = $this->clipManager->updateDcp(
                $arrayRequest);
                break;
            case 'delete_dcp':        $arrayResponse = $this->clipManager->deleteDcp(
                $arrayRequest);
```

```

        break;
    ...
    ...
    ...
    default:
        $arrayResponse = ['status' => 'ERR'];
        break;
    }
    $body = Json::encode($arrayResponse);
    $this->template->body = $body;
} catch (JsonException $ex) {
    $this->template->outputJson = Json::encode(["status" => "ERR"], Json::FORCE_ARRAY);
}
}
}

```

Výpis 5: Zpracování požadavku v souboru MainPresenter.php

6.3 Databázový model

Databázový model je základním pro všechny dále použité modely, které jsou jeho potomky. Abstraktní třída obsahuje jedinou metodu a tou je konstruktor. V něm je získáno připojení k databázi (*Context*), které je dále dostupné pro všechny jeho potomky. V Nette je pro práci s databází třída *DatabaseCore*, která umožňuje pracovat s databází v základním SQL režimu a nabízí všechny dostupné možnosti, a třída *DatabaseExplorer*, která umožní pracovat v programátorsky přívětivém režimu ovšem za cenu pouze jednoduchých dotazů.

```

abstract class DatabaseManager{
    use SmartObject;

    protected $database;

    public function __construct(Context $database){
        $this->database = $database;
    }
}

```

Výpis 6: Získání připojení k databázi v modelu DatabaseManager.php

Aby bylo možné se k databázi připojit, je nutné nastavit připojení. To se provádí v souboru *config.neon*. Zde nejprve uvedeme společnou sekci, kterou je *database*. Do ní poté vložíme klíč *dsn*, který předává parametr pro připojení PHP službě PDO, se kterou framework Nette pracuje. *Dsn* obsahuje údaje u typu databázového systému, jeho adresu a název databáze. Dalšími klíči

jsou uživatelské jméno a heslo, které se ve skutečnosti pochopitelně liší od toho uvedeného ve výpisu níže. Sekce obsahuje ještě parametry, které upřesňují typ spojení.

```
database:
  dsn: 'mysql:host=databaze.provider.com;dbname=nazev_databaze'
  user: uzivatelске_jmeno
  password: heslo
  options:
    lazy: yes
```

Výpis 7: Konfigurace připojení k databázi v souboru config.neon

6.4 Model pro manipulaci s DCP

Samotný model obsahuje několik metod, které již přímo provádějí operace s daty nad databází.

6.4.1 Vložení a úprava DCP

Vložení nového DCP se provádí metodou *insertDcp*, úprava pak metodou *updateDcp*. Jako vstupní argument obou metod je předáno asociativní pole, které vzniklo v presenteru rozbalením JSON příkazu. Nejprve je odstraněna položka obsahující příkaz. Poté je volána instance databáze, která se získala zděděním od databázového modelu. Té je předán název tabulky a samotná metoda *insert*, která v argumentu předává asociativní pole se všemi daty k vložení do databáze. U metody *updateDcp* je navíc předán také index záznamu, který se má aktualizovat. V případě, že vložení proběhne úspěšně, vrátí metoda kladnou odpověď pro klienta, v případě vyhození výjimky pak zápornou.

```
public function insertDcp($arrayRequest){
    try {
        unset ($arrayRequest['command']);
        $this->database->table(self::TABLE_DCP)->insert($arrayRequest);
        return ["status" => "OK"];
    }catch(\Exception $e) {
        return ["status" => "ERR"];
    }
}

public function updateDcp($arrayRequest){
    try {
        unset ($arrayRequest['command']);
        $this->database->table(self::TABLE_DCP)->where(self::COLUMN_DCP_ID, $arrayRequest[
            self::COLUMN_DCP_ID])->update($arrayRequest);
        return ["status" => "OK"];
    }
}
```

```

    }catch(Exception $e) {
        return ["status" => "ERR"];
    }
}

```

Výpis 8: Vložení a úprava DCP v modelu DcpManager.php

6.4.2 Odstranění DCP

Pro smazání záznamu DCP jsou dvě metody. Ta první funguje obdobně jako vložení nebo úprava. Pouze se zde nepředávají žádná data, ale jen index záznamu k odstranění. Metoda z databáze vymaže pouze DCP. V případě, že jsou k němu vázány nějaké KDM, zůstávají v databázi, čímž ale navždy ztrácejí referenci na konkrétní DCP.

```

public function deleteDcp($arrayRequest){
    try {
        unset ($arrayRequest['command']);
        $this->database->table(self::TABLE_DCP)->where(self::COLUMN_DCP_ID, $arrayRequest[
            self::COLUMN_DCP_ID])->delete();
        return ["status" => "OK"];
    }catch(\Exception $ex) {
        return ["status" => "ERR"];
    }
}

```

Výpis 9: Odstranění DCP v modelu DcpManager.php

Pokud si uživatel nepřeje po odstranění DCP zachovat ani KDM, musí použít následující metodu. Ta volá metodu předešlou a zároveň maže všechny záznamy z tabulky kdm, kde se v cizím klíči vyskytuje index odstraněného DCP.

```

public function deleteDcpWithKdms($arrayRequest){
    try {
        unset ($arrayRequest['command']);
        $this->deleteDcp($arrayRequest);
        $this->database->table(self::TABLE_KDM)->where(self::COLUMN_DCP_ID, $arrayRequest[
            self::COLUMN_DCP_ID])->delete();
        return ["status" => "OK"];
    }catch (Exception $e) {
        return ["status" => "ERR"];
    }
}

```

Výpis 10: Odstranění DCP včetně přiřazených KDM v modelu DcpManager.php

6.4.3 Výběr DCP

Pro všechny předchozí metody bylo použito třídy *DatabaseExplorer*, která umožňovala jednoduché operace s daty bez nutnosti psát SQL příkazy. Pro výběr zejména kvůli řazení již není možné tuto třídu použít a je nutné všechny SQL příkazy napsat ručně.

V metodě se nejprve odstraní položka příkazu a poté se ve struktuře switch společně s podmínkami if a else určí, podle čeho a jak mají být vybrané záznamy seřazeny.

```
public function selectDcps($arrayRequest){
    try {
        unset ($arrayRequest['command']);
        switch ($arrayRequest['sort_by']){
            case 'cpl':{
                if($arrayRequest['sort_up_down'] == 'ASC')
                    $sqlSort = 'ORDER BY dcp.dc_full_cpl_name ASC';
                else
                    $sqlSort = 'ORDER BY dcp.dc_full_cpl_name DESC';
            }break;
            ...
            ...
            ...
            default:
                $sqlSort = 'ORDER BY dcp.dc_full_cpl_name ASC';
            break;
        }
        $sqlBase = 'SELECT *
                    FROM dcp
                    ';

        $records = $this->database->fetchAll($sqlBase.$sqlSort);
    }
```

Výpis 11: Výběr a seřazení DCP v modelu DcpManager.php

Výsledkem je řetězec, který se připojí ke svému základu a společně jsou poslány jako SQL příkaz do databáze. Může vypadat například takto:

```
SELECT *
FROM dcp
ORDER BY dcp.dc_full_cpl_name DESC
```

Výpis 12: SQL příkaz pro výběr všech DCP z tabulky dcp seřazených sestupně podle názvu CPL

Dále je iterováno nad získanými záznamy a hodnoty obsahující čas a datum jsou správně naformátovány. Metoda v případě úspěchu vrací záznamy ve formě asociativního pole, v případě neúspěchu pak příznak chyby.

```

        $newDateTimeFormat = $record['dc_date_of_manufacture'];
        $record['dc_date_of_manufacture'] = $newDateTimeFormat->format('Y-m-d H:i:s');
    }
    if($record['dc_last_modify']){
        $newDateTimeFormat = $record['dc_last_modify'];
        $record['dc_last_modify'] = $newDateTimeFormat->format('Y-m-d H:i:s');
    }
}
return $records;
}catch(\Exception $ex){
    return ["status" => "ERR"];
}
}
}

```

Výpis 13: Formátování dat a jejich navrácení do presenteru v modelu DcpManager.php

6.5 Model pro manipulaci s KDM

Podobně jako předchozí popsaný model obsahuje i tento několik metod, které provádějí operaci s daty nad databází.

6.5.1 Vložení a úprava KDM

Metody *insertKdm* a *updateKdm* fungují analogicky k metodám pro vkládání a úpravu DCP. Opět je jako argument předáno asociativní pole vzniklé v presenteru rozbalením JSON příkazu. V čem se metody liší je pouze jiný název tabulky a sloupců. V případě, že vložení proběhne úspěšně, vrátí metoda rovněž kladnou odpověď pro klienta, v případě vyhození výjimky pak zápornou.

```

public function insertKdm($arrayRequest){
    try {
        unset($arrayRequest['command']);
        $this->database->table(self::TABLE_NAME)->insert($arrayRequest);
        return ["status" => "OK"];
    }
    catch(\Exception $e) {
        return ["status" => "ERR"];
    }
}

```

```

public function updateKdm($arrayRequest){
    try {
        unset ($arrayRequest['command']);
        $this->database->table(self::TABLE_NAME)->where(self::COLUMN_KDM_ID, $arrayRequest[
            self::COLUMN_KDM_ID])->update($arrayRequest);
        return ["status" => "OK"];
    }catch(\Exception $e){
        return ["status" => "ERR"];
    }
}

```

Výpis 14: Vložení a úprava KDM v modelu KdmManager.php

6.5.2 Odstranění KDM

Pro odstranění KDM z databáze existuje jedna metoda a to *deleteKdm*. Ze vstupního argumentu je vytažen index KDM, které je následně z databáze smazáno.

```

public function deleteKdm($arrayRequest){
    try {
        unset ($arrayRequest['command']);
        $this->database->table(self::TABLE_NAME)->where(self::COLUMN_KDM_ID, $arrayRequest[
            self::COLUMN_KDM_ID])->delete();
        return ["status" => "OK"];
    }catch (Exception $e) {
        return ["status" => "ERR"];
    }
}

```

Výpis 15: Odstranění KDM v modelu KdmManager.php

6.5.3 Výběr KDM

SQL příkaz pro výběr KDM je složitější než pro výběr DCP. Jelikož by uživateli samotná data uložená v tabulce KDM neposkytla dostatečné informace, je při výběru připojeno několik sloupců z tabulky dcp. Tento složitější dotaz s joinem na jinou tabulku již nelze podobně jako v případě vložení DCP zapsat pomocí třídy *DatabaseExplorer*, ale bylo nutné použít třídu *DatabaseCore*.

V metodě se opět nejprve odstraní položka příkazu a poté se ve struktuře switch společně s podmínkami if a else určí, podle čeho a jak mají být vybrané záznamy seřazeny.

```

public function selectKdms($arrayRequest){
    try {
        unset ($arrayRequest['command']);

```

```

switch ($arrayRequest['sort_by']){
    case 'kdm_name':{
        if($arrayRequest['sort_up_down'] == 'ASC')
            $sqlSort = 'ORDER BY kdm.kd_name ASC';
        else
            $sqlSort = 'ORDER BY kdm.kd_name DESC';
        }break;

    case 'cpl_name':{
        if($arrayRequest['sort_up_down'] == 'ASC')
            $sqlSort = 'ORDER BY dcp.dc_full_cpl_name ASC';
        else
            $sqlSort = 'ORDER BY dcp.dc_full_cpl_name DESC';
        }break;

    ...

    ...

    ...

    default:{
        $sqlSort = 'ORDER BY dcp.dc_full_cpl_name ASC';
    }break;
}

```

Výpis 16: Výběr a seřazení KDM v modelu KdmManager.php

Výsledkem je opět řetězec, který se připojí ke svému základu a stejně jako v případě výběru DCP je výsledný řetězec jako SQL příkaz poslán do databáze, který může vypadat například takto:

```

SELECT kdm.*,
dcp.dc_full_cpl_name,
dcp.dc_czech_name,
dcp.dc_english_name,
dcp.dc_original_name
FROM kdm
LEFT JOIN dcp
ON kdm.dc_id = dcp.dc_id
ORDER BY kdm.kd_name DESC

```

Výpis 17: SQL příkaz pro výběr všech KDM z tabulky kdm seřazených sestupně podle názvu CPL

Příkaz vybere požadované sloupce z tabulky *kdm* a připojí k nim záznamy z požadovaných sloupců z tabulky *dcp*. Spojení tabulek je realizováno příkazem LEFT JOIN. Ten zajistí, že jsou záznamům z tabulky uvedeného před příkazem (vlevo) přiřazeny záznamy z tabulky za příkazem, a to i v případě, že neexistuje shoda s tabulkou *dcp*. Tabulky se spojují cizím klíčem *dc_id* nazvaným shodně v obou tabulkách. Takový případ může nastat, pokud vymažeme pouze DCP, ke kterému zůstaly v databázi KDM, jenž mají právě na toto DCP referenci. V případě použití INNER JOIN nebo RIGHT JOIN by záznamy nebyly vybrány vůbec, pro uživatele by byly navždy již neviditelné, ale v databázi by nadále zůstaly.

Je-li proveden výběr z databáze, je následně jako v případě výběru DCP iterováno nad získanými záznamy, a hodnoty obsahující čas a datum jsou správně naformátovány. Metoda pro výběr KDM v případě úspěchu vrací záznamy ve formě asociativního pole, v případě neúspěchu pak příznak chyby.

```
foreach ($records as $record){
    if($record['kd_started_validity']){
        $newDateTimeFormat = $record['kd_started_validity'];
        $record['kd_started_validity'] = $newDateTimeFormat->format('Y-m-d H:i:s');
    }
    ...
    ...
    ...
}
return $records;
}catch (\Exception $ex) {
    return ["status" => "ERR"];
}
}
```

Výpis 18: Formátování dat a jejich navrácení do presenteru v modelu KdmManager.php

Pro výběr KDM z databáze existuje ještě jedna jednodušší metoda, která vrací méně dat a používá se k zobrazení KDM vázaných ke konkrétnímu DCP. Metoda se nazývá *selectKdm-sWithDcp* a jejím argumentem je opět asociativní pole. Stejně jako výše popsaná metoda vrací záznamy v podobě asociativního pole s naformátovanými časovými údaji. Jen již není možné zvolit řazení, které je nastaveno pevně na vzestupné podle data začátku platnosti. SQL příkaz je poté doplněn o klauzuli WHERE, která vybírá pouze ty záznamy, kde je jako cizí klíč nastaven index příslušného DCP.

6.6 Model pro manipulaci s uživateli

Pro práci s uživateli je v aplikačním serveru vytvořen samostatný model. Ten umožňuje získat informace o nich a ověřit (autentifikovat) uživatele pro přihlášení v samotné mobilní aplikaci.

6.6.1 Výběr uživatelů

Pro výběr všech uživatelů z databáze existuje metoda *getUser*. Ta vybere z databáze všechny uživatele se všemi uloženými parametry. Nad každým záznamem je následně iterováno a je z něj odstraněn otisk přihlašovacího hesla a naformátován správně časový údaj o poslední změně záznamu. Všechny záznamy jsou poté předány do presenteru, který je vrací aplikaci.

```
public function getUsers($arrayRequest){
    try {
        $records = $this->database->fetchAll(' SELECT *
                                           FROM      user
                                           ORDER BY  user.us_surname ASC');

        foreach ($records as $record){
            unset($record['us_hash_password']);
            if($record['us_last_modify']){
                $newDateTimeFormat = $record['us_last_modify'];
                $record['us_last_modify'] = $newDateTimeFormat->format('Y-m-d H:i:s');
            }
        }

        return $records;
    }catch(\Exception $ex){
        return ["status" => "UserManager ERROR"];
    }
}
```

Výpis 19: Výběr uživatelů a jejich navrácení do presenteru v modelu UserManager.php

6.6.2 Autentifikace uživatele

Pro autentifikaci (přihlášení) uživatele existuje metoda *authUser*. Dle uživatelského indexu předaného v argumentu metody je vytažen otisk hesla z databáze získán kryptovací metodou BCrypt. Tento otisk je porovnán se zasláným skutečným heslem z aplikace. V případě shody vrací server kladnou odpověď. V případě selhání zápornou, čímž mobilní aplikace může vyzvat uživatele k opakovanému přihlášení.

```
public function authUser(array $arrayRequest){
    try{
        $record = $this->database->fetch(' SELECT  user.us_hash_password
                                           FROM      user
                                           WHERE     user.us_id = ?', $arrayRequest['us_id']);
```

```
if(Passwords::verify($arrayRequest['us_plain_password'], $record['us_hash_password'])
){
    unset($record['us_hash_password']);
    return ["status" => "OK"];
}
else{
    return ["status" => "Auth ERROR"];
}
}catch(\Exception $ex){
    return ["status" => "UserManager ERROR"];
}
}
```

Výpis 20: Autentifikace uživatele a ověření hesla v databázi v modelu UserManager.php

Kompletní zdrojový kód aplikačního serveru je obsažen v příloze D.2

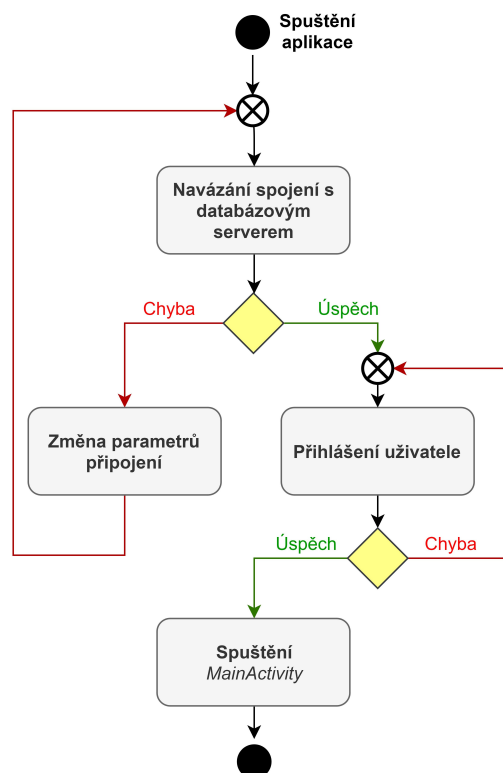
7 Struktura navržené aplikace

Aplikace byla napsána v jazyce Java. Jako vývojové prostředí bylo použito oficiální IDE založené na IntelliJ® Platform Android Studio verze 3.2 (později verze 3.3). Aplikace je navržena pro zařízení s API 19, které podporuje zařízení se systémem Android verze 4.4 a vyšší. Aplikace je tímto kompatibilní s 95,3 % provozovaných zařízení [26]. Pro komunikaci prostřednictvím sítě je využito vestavěného bezdrátového adaptéru zařízení. Komunikace probíhá na úrovni aplikační vrstvy za pomoci balíku *com.android.volley*, který zajišťuje zasílání HTTP a HTTPS požadavků, včetně zpracování následné odpovědi serveru.

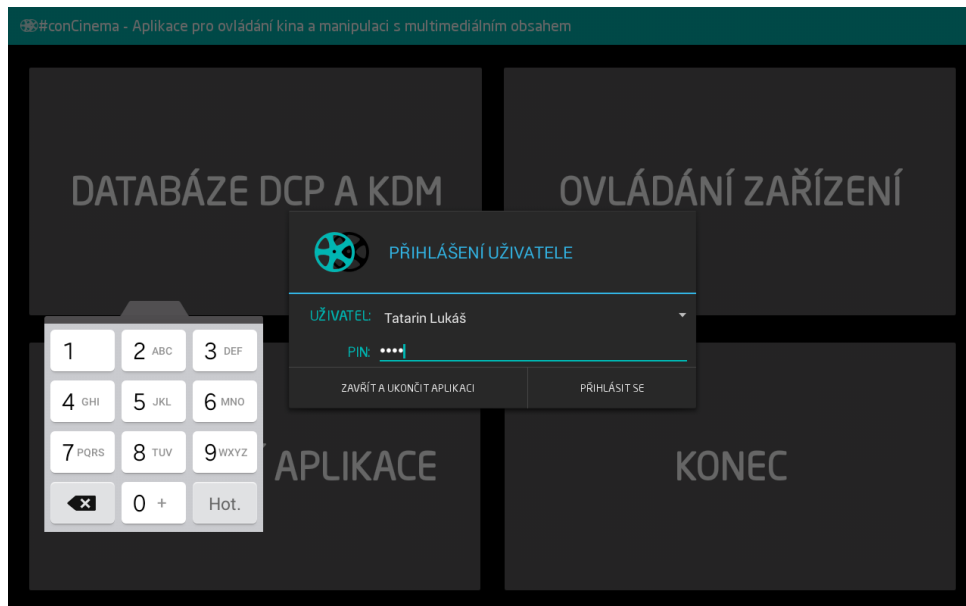
Celá aplikace je vzhledem k množství ovládacích prvků a zajištění komfortu ovládání navržena na zařízení s větší velikostí obrazovky. Ideální jsou větší mobilní telefony nebo tablety. Orientace celé aplikace je pevně stanovena na šířku obrazovky. Každá aktivita aplikace je vždy přes celou plochu obrazovky (fullscreen). V horní části se nachází toolbar, který informuje uživatele o dané aktivitě. V toolbaru se nachází také logo aplikace.

7.1 Spuštění aplikace a přihlášení

Po spuštění aplikace se otevře hlavní aktivita *MainActivity*, která testuje nastavené připojení k aplikačnímu (databázovému) serveru. Na server se posílá jednoduchý a krátký příkaz. Účelem tohoto testu je zjištění, zda-li je adresa serveru správně nastavena. Zároveň se tím server "probouzí" a navazuje se s ním spojení na úrovni síťové vrstvy. Toto spojení je nadále udržováno, čímž je následná komunikace rychlejší. Pokud se nepodaří se serverem navázat spojení, vyskočí možnost editovat jeho adresu. Jakmile je úspěšně provedeno spojení s databázovým serverem, je uživatel vyzván k výběru uživatelského jména a zadání přihlašovacího číselného pinu. Po přihlášení uživatele je naplno spuštěna tato aktivita. Proces navazování spojení a přihlášení se totiž děje v dialogovém okně, které hlavní aktivitu překrývá. Proces je znázorněn na následujícím diagramu a jako video-ukázka C.1 v příloze



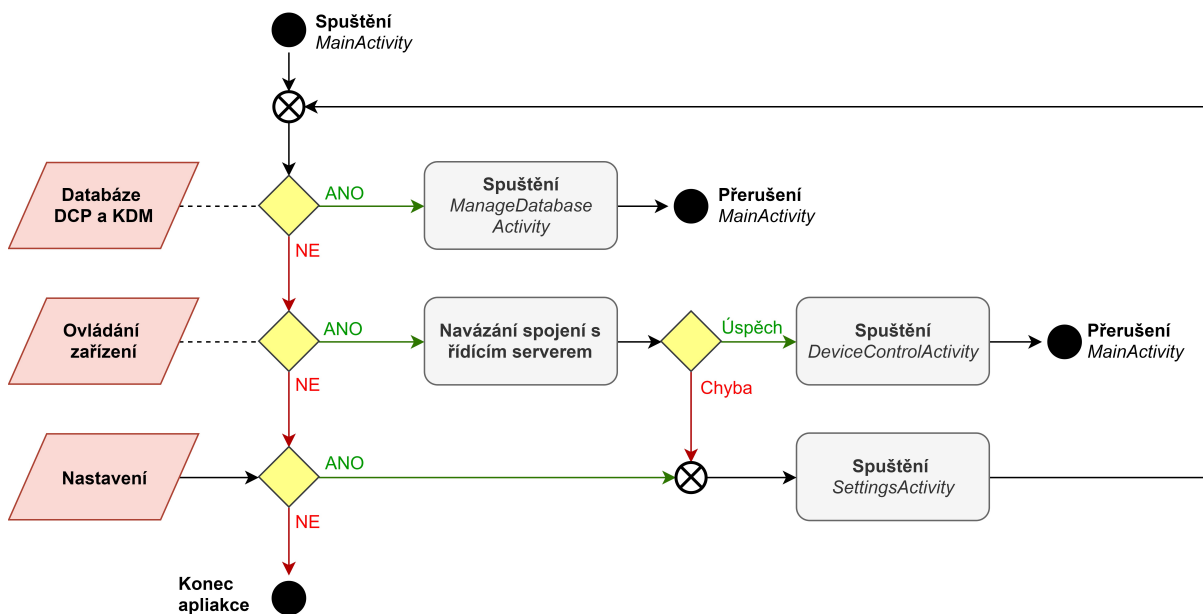
Obrázek 18: Aktivita diagram znázorňující spouštění aplikace a přihlášení uživatele, zdroj: autor



Obrázek 19: Snímek obrazovky s dialogovým oknem pro přihlášení, zdroj: autor

7.2 Hlavní aktivita

Hlavní aktivita funguje jako navigace. Nacházejí se zde čtyři tlačítka, která umožňují přihlášenému uživateli výběr, zda-li chce manipulovat s databází DCP a KDM, ovládat bezdrátově zařízení, změnit nastavení aplikace nebo celou aplikaci ukončit. V případě, že se rozhodne pracovat se záznamy v databázi, spustí se aktivita *ManageDatabaseActivity*. Pro vzdálené ovládání zařízení se spouští aktivita *DeviceControlActivity* s testem spojení na komunikační server. V případě, že je potřeba změnit nastavení aplikace, spustí se aktivita *SettingsActivity*, kde uživatel změní uložené preference. Z hlavní aktivity je rovněž možné celou aplikaci ukončit.



Obrázek 20: Aktivita diagram znázorňující návaznost dalších aktivit dle výběru akce, zdroj: autor

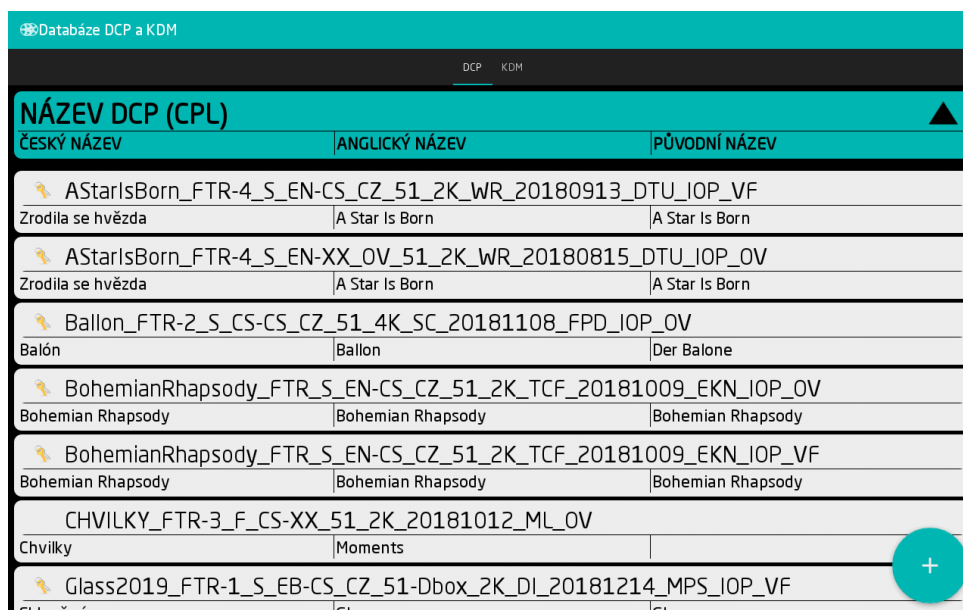
7.3 Prohlížení a editace DCP a KDM

S DCP a KDM uloženými v databázi se pracuje v aktivitě s názvem *ManageDatabaseActivity*. Samotná aktivita načítá do svého kontejneru jeden ze dvou fragmentů. Mezi fragmentem *DcpFragment* a *KdmFragment* lze přepínat swipem doleva, respektive doprava. Aktivní fragment je zobrazen také jako záložka v toolbaru umístěném pod hlavním toolbarem. Kliknutím na záložku lze rovněž fragment přepnout.

7.3.1 *DcpFragment*

Prvním a výchozím fragmentem při spuštění aktivity je *DcpFragment*. Součástí fragmentu je záhlaví seznamu, který zobrazuje výpis DCP uložených v databázi. Samotný seznam (výpis) je realizován pomocí *ListView*. Výpis je rolovací, pokud obsahuje více položek, než se vejde na obrazovku lze swipem nahoru respektive dolů zobrazit další položky. Záhlaví je ukotveno stabilně

a během rolování seznamem se jeho pozice nemění. Součástí fragmentu je také plovoucí tlačítko, kterým lze přidat do databáze nové DCP.



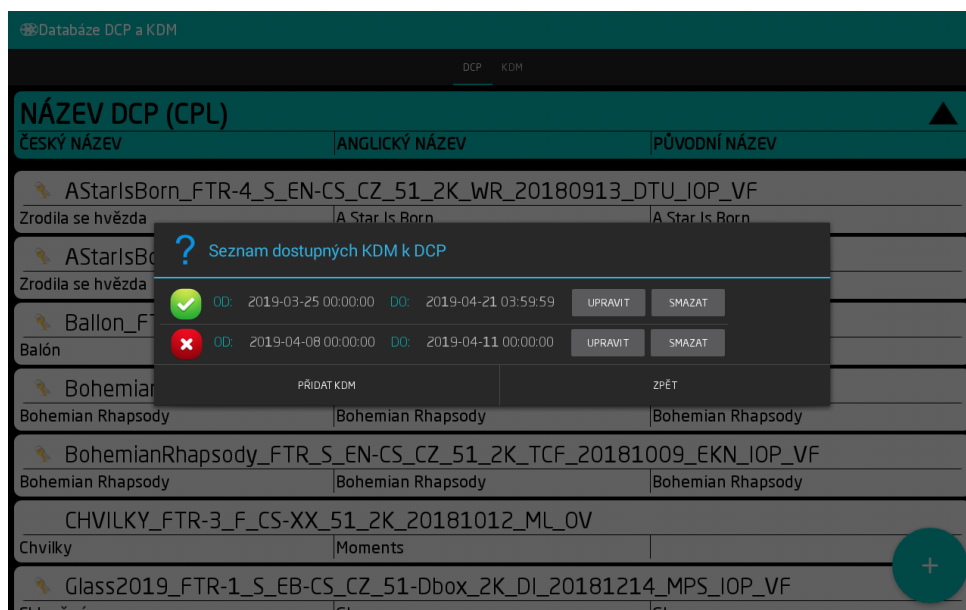
Databáze DCP a KDM		
DCP KDM		
NÁZEV DCP (CPL)		
ČESKÝ NÁZEV	ANGLICKÝ NÁZEV	PŮVODNÍ NÁZEV
AStarIsBorn_FTR-4_S_EN-CS_CZ_51_2K_WR_20180913_DTU_IOP_VF		
Zrodila se hvězda	A Star Is Born	A Star Is Born
AStarIsBorn_FTR-4_S_EN-XX_OV_51_2K_WR_20180815_DTU_IOP_OV		
Zrodila se hvězda	A Star Is Born	A Star Is Born
Ballon_FTR-2_S_CS-CS_CZ_51_4K_SC_20181108_FPD_IOP_OV		
Balón	Ballon	Der Balone
BohemianRhapsody_FTR_S_EN-CS_CZ_51_2K_TCF_20181009_EKN_IOP_OV		
Bohemian Rhapsody	Bohemian Rhapsody	Bohemian Rhapsody
BohemianRhapsody_FTR_S_EN-CS_CZ_51_2K_TCF_20181009_EKN_IOP_VF		
Bohemian Rhapsody	Bohemian Rhapsody	Bohemian Rhapsody
CHVILKY_FTR-3_F_CS-XX_51_2K_20181012_ML_OV		
Chvilky	Moments	
Glass2019_FTR-1_S_EB-CS_CZ_51-Dbox_2K_DI_20181214_MPS_IOP_VF		
Glasnost	Letos	Letos

Obrázek 21: Snímek obrazovky aktivity ManageDatabaseActivity s fragmentem DcpFragment, zdroj: autor

V záhlaví se nachází legenda k seznamu. Je možné kliknutím na jednotlivé nadpisy seřadit položky vzestupně nebo sestupně. Aktuální řazení je indikováno černými trojúhelníkovými značkami, které mění svou orientaci.

Každá položka v seznamu obsahuje ikonku indikující, zda-li je dané DCP chráněno pomocí KDM nebo ne. Dále je zde celý CPL název DCP a také česká, anglická a původní verze názvu DCP. Tyto vlastnosti jsou nejdůležitější pro uživatele, a proto jsou uvedeny u každé položky seznamu.

Krátkým stiskem položky v seznamu, jenž odkazuje na DCP chráněné pomocí KDM (ikona klíče) se objeví dialogové okno, které obsahuje výpis všech KDM přiřazených k danému DCP. Zde je možné jednotlivé KDM upravit nebo smazat. V případě, že je zvolena možnost úpravy, otevře se aktivita *UpdateKdmActivity*, v případě zvolení možnosti smazat KDM, je dialog překryt ještě jedním dialogem, který požaduje potvrzení záměru nenávratně KDM odstranit z databáze. Původní dialog, který se aktivuje krátkým stiskem nabízí také přidat nové KDM k DCP. Pokud není DCP chráněno KDM, nelze žádné KDM vyhledat a ani je zobrazit. Místo dialogu je tak uživatel informován hláškou v podobě vyskakovacího *Toastu*, že DCP není chráněno pomocí KDM.



Obrázek 22: Dialogové okno s výpisem KDM přiřazených k danému DCP, zdroj: autor

Dlouhým stiskem položky v seznamu je aktivováno dialogové okno, které zobrazí veškeré dostupné vlastnosti daného DCP. Uživatel může přímo z dialogového okna vyvolat i další akce. Odstranění DCP z databáze se provede tlačítkem pro smazání, po jehož stisku je dialogové okno opět překryto dalším oknem, kde se po uživateli požaduje potvrzení nevratné operace. Je-li DCP označeno jako chráněné, lze navíc zvolit z možností odstranit včetně KDM nebo přiřazené KDM v databázi prozatím ponechat. Druhou možností je upravit vlastnosti stávajícího záznamu v databázi.



Obrázek 23: Dialogové okno s výpisem vlastností daného DCP, zdroj: autor

Prohlížení záznamu DCP je zaznamenáno na video-ukázce v příloze C.2, odstranění pak na video-ukázce v příloze C.8.

7.3.2 *InsertDcpActivity* a *UpdateDcpActivity*

Vyvoláním akce pro úpravu se spustí aktivita *UpdateDcpActivity*, která umožňuje před vyplněním jednotlivých editačních prvků upravovat vlastnosti DCP. Po dokončení úprav a stisku tlačítka pro aktualizaci údajů, aktivita data zpracuje a pošle na server.

Podobným způsobem funguje i aktivita *InsertDcpActivity*, která využívá stejný layout jako aktivita pro editaci. Spouští se kliknutím na plovoucí tlačítko v *DcpFragment*. Po vyplnění editačních polí jsou data zpracována a odeslána na server, který automaticky přiřadí index nového záznamu. Ten není nutné předem zjišťovat.

Upravit stávající DCP v databázi

CPL:
CHVILKY_FTR-3_F_CS-XX_51_2K_20181012_ML_OV

ROZPOZNĚJ VLASTNOSTI

Český název:	Anglický název	Původní název:
Chvilky	Moments	

Typ obsahu:	Verze obsahu:	Prostorová verze:	FPS:	Formát obrazu:	Jazyk zvuku:
Feature / Film (FTR)	3	2D	24	FLAT (F)	CS - Čeština a Sloven.
Jazyk titulků:	Lokalizace:	Zvukový formát:	Rozlišení obrazu:	DCI standard:	DCI verze:
XX - Žádný	bez specifikované lok.	5.1 - Dolby DIGITAL	2K	IOP - Interop	OV - original version
Datum výroby DCP:	Prostorový zvuk:				
2018-10-10	Žádná	<input type="radio"/> HI stopa	<input type="radio"/> VI stopa	<input type="radio"/> SL stopa	<input type="radio"/> DBOX
<input type="radio"/> Chráničeno KDM					

Poznámka:

Oceněn Českým lvem 2018

UPRAVIT STÁVAJÍCÍ DCP V DATABÁZI

Obrázek 24: Aktivita pro úpravu stávajícího DCP, zdroj: autor

Vložení nového záznamu DCP je zaznamenáno na video-ukázce v příloze C.4, úprava stávajícího záznamu pak na video-ukázce v příloze C.6.

7.3.3 *KdmFragment*

Druhým fragmentem dostupným v *ManageDatabaseActivity* je *KdmFragment*. Součástí fragmentu je stejně jako v případě *DcpFragment* záhlaví seznamu, který zobrazuje tentokrát výpis KDM uložených v databázi. Seznam je opět realizován pomocí *ListView* a umožňuje zobrazit více položek, než se vejde na obrazovku. Záhlaví je ukotveno stabilně a během rolování seznamem se jeho pozice nemění. Součástí fragmentu je také plovoucí tlačítko, kterým lze přidat do databáze nové KDM.

Databáze DCP a KDM			
DCP KDM			
NÁZEV KDM	NÁZEV DCP (CPL)	ZAČÁTEK PLATNOSTI	KONEC PLATNOSTI
✗ Ballon_FTR-2_S_CS-CS_CZ_51_4K_SC_20181108_FPD_IOP_OV	Ballon_FTR-2_S_CS-CS_CZ_51_4K_SC_20181108_FPD_IOP_OV	2019-04-22 00:00:00	2019-04-28 23:59:59
✗ KDM generated for A31177	BohemianRhapsody_FTR_S_EN-CS_CZ_51_2K_TCF_20181009_EKN_....	2019-04-08 00:00:00	2019-04-11 00:00:00
✓ KDM generated for A39911	AStarIsBorn_FTR-4_S_EN-CS_CZ_51_2K_WR_20180913_DTU_IOP_....	2019-03-27 00:00:00	2019-04-28 23:59:00
✓ KDM generated for A39911	BohemianRhapsody_FTR_S_EN-CS_CZ_51_2K_TCF_20181009_EKN_....	2019-03-25 00:00:00	2019-04-21 03:59:59
✗ KDM_Kino-Orion_Hradec_nad_Moravici	ZLATY-PODRAZ_FTR-1_S_CS-CS_INT-TL_51-XX_4K_20181012_UPP_....	2019-04-19 00:03:00	2019-04-22 00:00:59

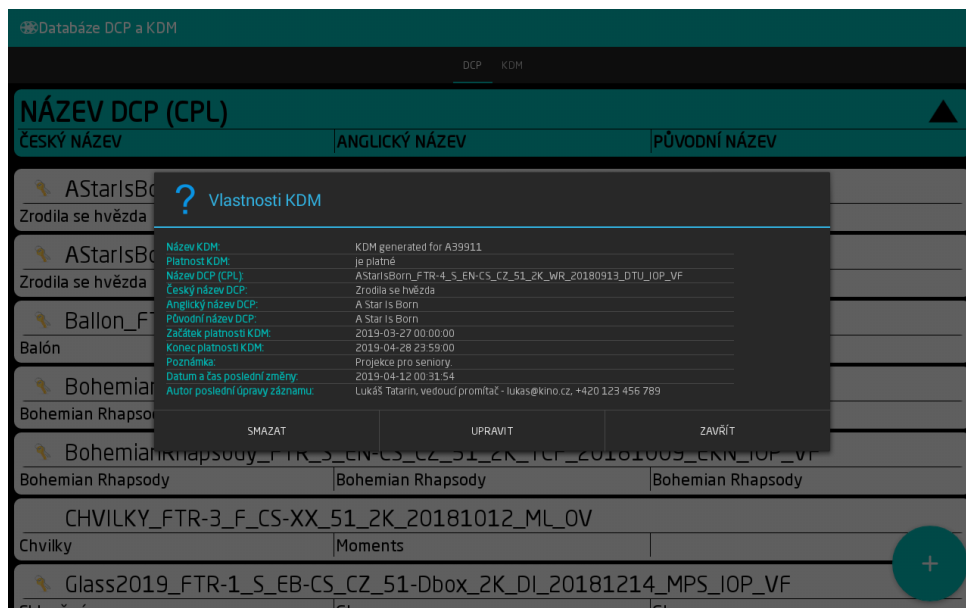
Obrázek 25: Snímek obrazovky aktivity ManageDatabaseActivity s fragmentem KdmFragment, zdroj: autor

V záhlaví se nachází legenda k seznamu. Je možné kliknutím na jednotlivé nadpisy seřadit položky vzestupně nebo sestupně. Aktuální řazení je indikováno černými trojúhelníkovými značkami, které mění svou orientaci.

Každá položka v seznamu obsahuje ikonku indikující, zda-li je dané KDM k současnému datu (včetně aktuálního času) platné nebo nikoliv. Dále je zobrazen název KDM tak, jak jej certifikační autorita během generování pojmenovala, název CPL, ke kterému se KDM váže a datum a čas začátku a konce platnosti. Tyto vlastnosti jsou pro operátora nejdůležitější, a proto jsou uvedeny u každé položky v seznamu.

Dlouhým stiskem položky v seznamu je aktivováno dialogové okno, které zobrazí veškeré dostupné vlastnosti daného KDM. Uživatel může přímo z dialogového okna vyvolat i další akce. Odstranění KDM z databáze se provede tlačítkem pro smazání, po jehož stisku je dialogové okno opět překryto dalším oknem, kde se po uživateli požaduje potvrzení nevratné operace. Druhou možností je upravit vlastnosti stávajícího záznamu v databázi.

Prohlížení záznamu KDM je zaznamenáno na video-ukázce v příloze C.3, odstranění pak na video-ukázce v příloze C.9.



Obrázek 26: Dialogové okno s výpisem vlastností daného KDM, zdroj: autor

7.3.4 *InsertKdmActivity* a *UpdateKdmActivity*

Vyvoláním akce pro úpravu se spustí aktivita *UpdateKdmActivity*, která umožňuje před vyplněním jednotlivých editačních prvků upravovat vlastnosti KDM. Jedinou vlastnost, kterou již nelze měnit, je DCP, ke kterému je KDM přiřazeno. Po dokončení úprav a stisku tlačítka pro aktualizaci údajů, aktivita data zpracuje a pošle na server.

Podobným způsobem funguje i aktivita *InsertKdmActivity*, která využívá stejný layout jako aktivita pro editaci. Spouští se kliknutím na plovoucí tlačítko v *KdmFragment*. Po vyplnění editačních polí jsou data zpracována a odeslána na server, který automaticky přiřadí index nového záznamu a index (cizí klíč) DCP, který je vybrán v aktivitě.

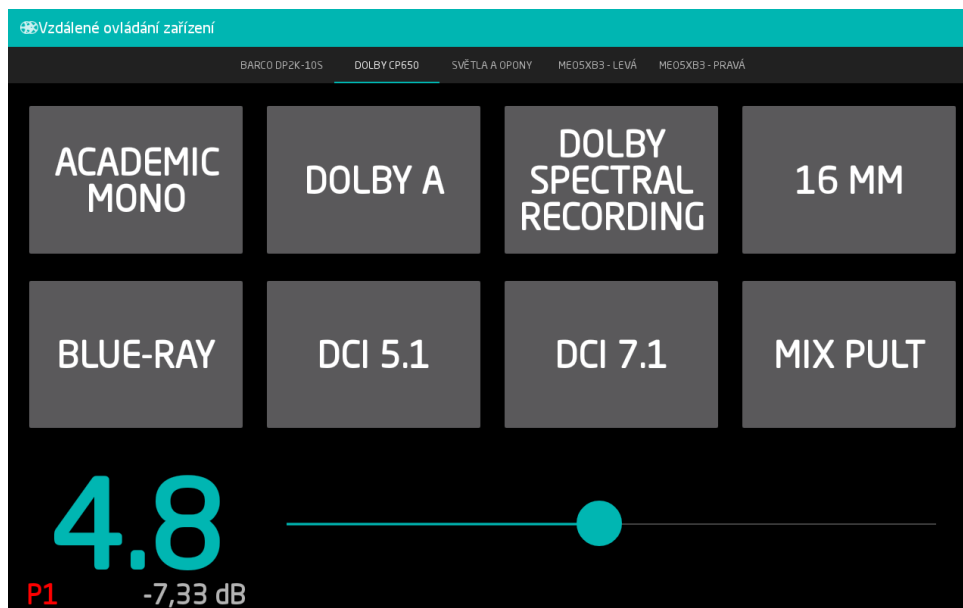
Obrázek 27: Aktivita pro úpravu stávajícího DCP, zdroj: autor

Vložení nového záznamu KDM je zaznamenáno na video-ukázce v příloze C.5, úprava stávajícího záznamu pak na video-ukázce v příloze C.7.

7.4 Ovládání zařízení

Ovládání zařízení se děje po spuštění *DeviceControllActivity* pomocí navigace v *MainActivity*. Součástí aktivity pro ovládání jednotlivých zařízení je několik fragmentů. V současné době jich je pět, avšak architektura aplikace umožňuje kdykoliv přidat další fragmenty. Mezi jednotlivými fragmenty lze přepínat swipem doleva, respektive doprava. Aktivní fragment je zobrazen také jako záložka v toolbaru umístěném pod hlavním toolbarem. Kliknutím na záložku lze rovněž fragment přepnout. Fragment *DigitalProjectorFragment* slouží k bezdrátovému ovládání digitálního DCI projektoru, fragment *SoundProcessorFragment* pak k ovládání zvukové Dolby jednotky. Fragment *PanelFragment* nahrazuje fyzický operátorský panel pro ovládání osvětlení, stmívačů, maskování a opon. Poslední dvojice fragmentů *AnalogProjectorLeftFragment* a *AnalogProjectorRightFragment* k obsluze 35mm kino-projektoru.

Fragmenty obsahují indikační a ovládací prvky. Aktivací každého prvku se odešle příkaz na komunikační server umístěný v lokální síti. Odpověď a úspěšné vykonání příkazu je indikováno pomocí indikačního prvku (např. *TextView*), kdy zobrazí nově nastavenou hodnotu, jakou může být například hlasitost, nebo změnou pozadí ovládacího prvku (např. *Button*), kdy se označí daná volba jako aktivní (přepínání zvukových formátů). Pokud stiskem tlačítka odesíláme příkaz, který nemá zpětnou vazbu, ale pouze spouští určitý proces (pohyb opon, dosažení úrovně osvětlení), je uživatel o úspěšném vykonání informován pomocí vyskakovací zprávy *Toast*.



Obrázek 28: Fragment pro ovládání instalované zvukové jednotky DOLBY CP650, zdroj: autor

Celá komunikace s řídicím serverem probíhá asynchronně. Pokud je určitá akce vykonána jiným způsobem (přímo na daném zařízení) než prostřednictvím aplikace, uživatel nemusí mít vizualizován aktuální stav, protože řídicí systém neposílá aplikaci informace, pokud k tomu není aplikací vyzván. Proto při každém spuštění fragmentu je načten celkový stav ovládaného zařízení a zároveň spuštěn na pozadí časovač, který se v pravidelných intervalech na aktuální stav dotazuje. Při přerušení fragmentu je pak tento časovač zastaven.

Ovládání zařízení je zaznamenáno na video-ukázce v příloze C.10.

7.5 Komunikace aplikace s databázovým a řídicím serverem

Jak již bylo zmíněno výše v popisu implementace aplikačního serveru, tak komunikace s tímto serverem, který provádí úkony spojené s databází probíhá pomocí protokolu HTTPS. Tím je přenos dat chráněn proti nechtěnému zneužití útokem typu MITM. Tedy útočníkem, který by mohl poslouchat v síti mezi klientem a serverem a snažit se nešifrovanou komunikaci rozpoznat. Podmínkou bezpečné komunikace je věrohodný certifikát, který zajišťuje asymetrickou výměnu klíče.

Pro komunikaci s řídicím serverem se používá protokolu HTTP, jelikož zabezpečená verze na řídicím serveru není podporována. Vzhledem k topologii instalace je předpoklad umístění řídicího serveru do zabezpečené a neveřejné části lokální sítě, kde se útok typu MITM nepředpokládá.

7.5.1 Komunikační API

Samotná data jsou předávána v serializačním formátu JSON. Každý jeden příkaz nebo vrácený záznam je ve tvaru JSON objektu. Pokud je třeba odeslat více příkazů nebo získat více záznamů

najednou, je třeba všechny JSON objekty příkazů či záznamů vložit do jednoho datového typu JSON pole.

Každý příkaz pro aplikační server obsahuje klíč *command*, který svou hodnotou specifikuje akci, jež aplikační server nad databází provede. Následně je dvojice doplněna o data, kde klíče představují jednotlivé názvy sloupců databáze (viz datový slovník) a hodnotu pak obsah buňky, která byla vyčtena, či má být vložena nebo upravena.

Na příkladu níže vidíme konstrukci JSON řetězce pro vložení nového DCP do databáze. Řetězec jako klíče obsahuje názvy všech sloupců tabulky *dcp*, do kterých chceme vložit data. Uprostřed řetězce je vložen příkaz.

```
{
  "dc_czech_name" : "Český název",
  "dc_english_name" : "Anglický název",
  "dc_original_name" : "Původní název",
  "dc_aspect_ratio" : 4,
  ...
  "command": "insert_dcp",
  ...
  "dc_encrypted" : 0
}
```

Výpis 21: Příklad JSON řetězce pro vložení nového DCP

Vrácené záznamy ve formátu JSON nesou klíče které opět představují jednotlivé názvy sloupců databáze a hodnoty pak obsahy buněk, které byly z databáze vyčteny.

Na příkladu níže vidíme konstrukci JSON řetězce obsahující několik záznamů KDM, kdy informaci o každém KDM nese jeden JSON objekt, který je součástí JSON pole.

```
[
  {
    "kd_id" : 49,
    "dc_id" : 36,
    "kd_name" : "KDM generated for A39911",
    "kd_started_validity" : "2019-01-2900:00:00",
    "kd_ended_validity" : "2019-01-30 23:59:00"
    "kd_note" : "Projekce pro seniory.",
    "kd_last_modify" : "2019-01-29 10:49:14",
    ...
  },
  {
    ...
    "kd_name" : "KDM generated for mps-newquay-wm1-arch ~ KDM for LE MD FM SM",
    "kd_started_validity" : "2019-01-25 00:00:01",
    ...
  }
]
```

```
"kd_ended_validity" : "2019-01-28 03:59:59",
"kd_note" : "Na poslední lednový víkend.",
"kd_last_modify":"2019-01-23 14:53:23"
}
]
```

Výpis 22: Příklad JSON řetězce navraceného serverem obsahující více záznamu KDM

Příkazy pro řídicí server mají obdobnou strukturu jako ty pro obsluhu databáze. Každý příkaz obsahuje klíč *device*, který specifikuje zařízení, které má být ovládáno. Následují klíče nastavení/přepnutí jednotlivých parametrů a hodnota, která má být nastavena. Pochopitelně pro každé zařízení se nastavitelné parametry liší.

```
{
  "device":"sound_processor",
  "fader":"045"
}
```

Výpis 23: Příklad JSON řetězce obsaženého v požadavku a odpovědi pro nastavení hlasitosti zvukové jednotky na DCI úroveň 4.0 (cca -8.33 dBC)

Server po nastavení odpoví stejným řetězcem, který indikuje, že nastavení proběhlo správně a posílá aktuální hodnoty. Ty se v drtivé většině případů neliší od hodnot zaslaných v požadavku. Výjimku může utvořit stav, kdy uživatel zadá nesprávnou hodnotu nebo hodnotu mimo nastavitelnou mez. Pak server vrátí skutečnou hodnotu, které se podařilo dosáhnout.

Pro zjištění aktuálního stavu je v případě zvukové jednotky využito následujícího příkazu. Příkaz je stejný pro všechna zařízení. Liší se ale v parametru klíče *device*. Tento příkaz je také v pravidelných intervalech odesílán časovačem běžícím na pozadí aplikace.

```
{
  "device" : "sound_processor",
  "command" : "get_all"
}
```

Výpis 24: Příklad JSON řetězce obsaženého v požadavku pro zjištění všech parametrů zvukové jednotky

Server v tomto případě neodpovídá stejným tvarem, ale vrací konkrétní hodnoty všech nastavených parametrů. Ty se opět liší dle použitého zařízení.

```
{
  "device" : "sound_processor",
  "fader": "045",
  "mute" : "0",
  "format_button" : "4",
}
```

```
"projector" : "1"
}
```

Výpis 25: Příklad JSON řetězce obsaženého v odpovědi pro zjištění všech parametrů zvukové jednotky

7.5.2 Odeslání a zpracování HTTP a HTTPS požadavků

O komunikaci se serverem pomocí protokolu HTTP nebo HTTPS se stará třída *MyHttpAdapter*, využívající balíku *com.android.volley*. Třídě se v konstruktoru předává *Context* dané aktivity, která volá její metody a také URL, které kromě doménového jména či IP adresy specifikuje také komunikační protokol (HTTP nebo HTTPS). Dále má třída implementované callbacky v podobě interfaců, které slouží k asynchronnímu zpracování odpovědi. Kód obsažený v callbacku se provede až po odpovědi serveru.

Jako příklad je uvedena implementace metody, která odesílá na server požadavek ve formátu JSON objektu, přičemž odpověď očekává ve stejném formátu. Nejprve je vytvořena instance ze třídy *JsonObjectRequest*, které při vytváření předáme metodu požadavku, kterou se serveru dotáže. V tomto případě je to typ POST. Dalším argumentem je URL s adresou serveru a samotná data ve formátu JSON objektu. Posledními dvěma argumenty jsou callbacky (listenery), které obsahují procedury vykonané po odpovědi serveru, případně při výskytu chyby v komunikaci nebo zpracování požadavku. V obou listenerech se volá další callback, který předává data dále a zároveň se ukončuje fronta s daným požadavkem. Požadavek se odesílá až po přidání vytvořené instance do fronty. Nutno zmínit, že níže uvedený kód se neprovádí chronologicky, tak jak je zapsán shora dolů, ale po obdržení odpovědi od serveru se skáče zpátky do listenerů definovaných při vytváření instance.

```
public void newJSONObjectRequest(JSONObject jsonBody, final JSONObjectCallback callback) {
    JsonObjectRequest jsonObjectRequest = new JsonObjectRequest(Request.Method.POST, url,
        (jsonBody == null) ? null : jsonBody,
        new Response.Listener<JSONObject>() {
            @Override
            public void onResponse(JSONObject response) {
                callback.onJSONObjectRequestSuccess(response);
                requestQueue.stop();
            }
        },
        new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error){
                if (error instanceof TimeoutError || error instanceof NoConnectionError)
```

```

        Toast.makeText(context, context.getString(R.string.error_network),
            Toast.LENGTH_SHORT).show();
    else if (error instanceof ServerError)
        Toast.makeText(context, context.getString(R.string.error_server), Toast
            .LENGTH_SHORT).show();
    callback.onJSONObjectRequestFailure(error);
    requestQueue.stop();
    }
    });
    this.requestQueue.add(jsonObjectRequest);
}

```

Výpis 26: Implementace odeslání a zpracování `JSONObjectRequest` požadavku v souboru `MyHttpAdapter.java`

Třída *MyHttpAdapter* dále obsahuje metody, které umí zpracovat odpovědi ve formátu JSON pole, přičemž data na server jsou poslána ve formátu JSON objektu. Třída *JsonObjectRequest* je potomkem třídy *StringRequest* rozšířenou o kontrolu celistvosti JSON řetězce. Oproti *StringRequest* umožňuje rozeznat nekompletní JSON nebo nekompatibilní tvar.

7.5.3 Uložení a manipulace s daty

Získaná data a informace o jednotlivých DCP a KDM se ukládají ve třídách *DcpClass* a *KdmClass*. Každá instance vytvořená z těchto tříd nese informace o právě jednom DCP či KDM. Třída ukládá vlastnosti v datovém typu *HashMap*, kde klíč každé vlastnosti je shodný s názvem sloupce v databázi a obsah buňky, která je průnikem daného záznamu a sloupce, je pak hodnotou přiřazenou ke klíči. Třída dále obsahuje property (getter a setter) pro čtení a zápis hodnot a metody pro zjištění platnosti DCP nebo KDM. Vytvoření instance je možné prázdným konstruktorem, který instanci nenaplní žádnými daty. Ta jsou do instance přidána po vyplnění dat uživatelem pomocí výše zmíněných properties. Druhému konstruktoru je v argumentu předán datový typ *JSONObject*. Nad daty je iterováno a automaticky jsou přidány do instance, kde je lze poté přecházet nebo přepsat. Typicky při vytvoření instance na základě dat získaných odpovědí serveru.

Obě třídy dále obsahují privátní konstruktor a veřejnou metodu po zabalení a následné rozbalení celé instance pomocí třídy *Parcel*. Díky tomu je možné instance předávat mezi aktivitami. Například získaná data jsou předána do jiné aktivity, která umožní uživateli jejich úpravu a následné uložení zpátky do databáze.

7.5.4 Vložení a výběr dat z databáze

Pro vložení a výběr dat z databáze slouží třída *MysqlQuery*. Obsahuje jednotlivé metody pro vložení, úpravu, mazání nebo výběr DCP či KDM. Každá z těchto metod využívá třídu *DcpClass*

nebo *KdmClass*. Buď jejich obsah využívá jako vstupní data pro vložení či editaci, nebo navrácí instanci těchto tříd jako odpověď získanou dotazem z databáze. Metody pak využívají třídu *MyHttpAdapter* sloužící právě ke komunikaci se serverem.

Metoda *selectDcps* slouží k výběru DCP z databáze, která vrací v callbacku generickou třídu *List* sestávající se z instancí třídy *DcpClass*. První dva argumenty funkce jsou řetězce, které určují, podle kterého sloupce a v jakém pořadí proběhne seřazení získaných záznamů. Následně je definován callback, jehož procedury se provádějí po obdržení odpovědi serveru. Po zavolání metody *selectDcps* se vytvoří nová instance třídy *MyHttpAdapter*, které se předá context aplikace a URL adresa serveru. Poté je z této instance volána metoda, která dle komunikačního API odešle na server řetězec v JSON tvaru, který požaduje výběr všech DCP seřazených dle vstupních argumentů metody *selectDcps*. Konstrukci JSON řetězce obstará privátní statická metoda *SELECT_DCPS*. V callbacku *onJSONArrayRequestSuccess* se předpokládá odpověď datového typu *JSONArray*. Následně je zjištěna délka přijatého pole a vytvořena již výše zmíněná generická třída *List* obsahující instance třídy *DcpClass*. Nad polem výsledku je iterováno tolikrát, jak velká je jeho délka, kdy při každém cyklu je vytvořena a přidána do proměnné *dcpsList* nová instance třídy *DcpClass*, která je vytvořena pomocí konstruktoru, který rozparsuje jeden *JSONObject* obsažený v odpovědi ve tvaru *JSONArray*. Pokud je délka pole (počet záznamů) roven například deseti, proběhne iterační cyklus právě desetkrát. Po skončení posledního cyklu je volán callback, jenž vrací *List* naplněný instancemi *DcpClass*.

```
public void selectDcps(String sortBy, String sortUpDown, final SelectDcpsCallback callback)
{
    MyHttpAdapter myHttpAdapter = new MyHttpAdapter(this.context, this.url);
    myHttpAdapter.newJSONArrayRequest(SELECT_DCPS(sortBy, sortUpDown),
        new MyHttpAdapter.JSONArrayCallback(){
            @Override
            public void onJSONArrayRequestSuccess(JSONArray result){
                List<DcpClass> dcpsList = new ArrayList<>();
                for (int i = 0; i < result.length(); i++){
                    try {
                        dcpsList.add(new DcpClass(result.getJSONObject(i)));
                    }catch (JSONException je){
                        Log.e("selectKdmsException", je.toString());
                    }
                }
                callback.onSelectDcpsSuccess(dcpsList);
            }
        });
}
```

Výpis 27: Implementace výběru DCP z databáze v souboru *MysqlQuery.java*

8 Testování aplikace

Výše popsaný způsob návrhu a implementace aplikace je plně funkční. Aplikace dokáže manipulovat s daty v databázi a komunikovat s řídicím serverem. Všechny funkcionality byly reálně odzkoušeny a odstraněny případné nedostatky.

Pro účel testování rychlosti odezvy aplikačního a řídicího serveru byla v aplikaci účelně naprogramována metoda zasílající HTTP a HTTPS požadavky. Rychlost odezvy pak byla změřena pomocí nástroje *Profiler*, který dovede změřit síťový provoz včetně objemu zaslaných dat.

8.1 Rychlost odezvy v komunikaci s databázovým serverem

Pro testování databázového serveru bylo vytvořeno fiktivní DCP. Posléze bylo takové DCP několikrát vloženo do databáze a změřena rychlost odezvy serveru na takový požadavek. Většina vlastností DCP je datového typu *enum*. Délka samotného JSON řetězce, který nese vlastnosti DCP se tak v jednotlivých případech příliš nemění.

Poté bylo několik záznamů stejného DCP z databáze vybráno a opět byla změřena rychlost odezvy. Naměřené hodnoty jsou obsaženy v tabulce. Pro úplnost dodám, že na serveru a v aplikaci byla vypnuta možnost cashování. Měření probíhalo prostřednictvím Wi-Fi připojení k internetu skrze telefonní linku CETIN o maximální rychlosti 20 Mbit/s a prostřednictvím mobilního datového připojení pomocí technologie LTE.

Tabulka 2: Srovnání rychlosti odezvy databázového serveru s připojením prostřednictvím mobilních dat

Počet záznamů	1	10	100	1000
Čas vložení záznamů [ms]	189 (1)	1145 (6)	2798(49)	69563 (8)
Čas výběru záznamů [ms]	360 (1)	241 (1)	248 (1)	487 (1)

Tabulka 3: Srovnání rychlosti odezvy databázového serveru s připojením prostřednictvím telefonní linky CETIN

Počet záznamů	1	10	100	1000
Čas vložení záznamů [ms]	145 (1)	1018 (5)	2598(21)	60227 (7)
Čas výběru záznamů [ms]	361 (1)	254 (1)	260 (1)	434 (1)

Z naměřených výsledků je patrné, že doba pro vložení záznamu je přímo úměrná počtu vložených záznamů. Je to dáno tím, že záznamy se nevkládají během jednoho požadavku, ale jako fronta za sebou. Aplikace si pro urychlení zpracování v případě většího množství záznamu otevřela hned několik socketů na serveru. Jejich počet je v každé buňce tabulky uveden v závorce. Případ, že by bylo okamžitě za sebou (zároveň) vkládáno několik záznamů je nepravděpodobný. Uživatel musí jednotlivá data nejprve vyplnit a poté je teprve může na databázový server odeslat.

V případě výběrů záznamů z databáze již není s rostoucím počtem záznamů nárůst doby odezvy tak enormní jako v případě jejich vložení. Stažení probíhá v jednom požadavku, a tím pádem i při jednom otevřeném socketu. Liší se jen velikost odpovědi v počtu vybraných záznamů.

Z měření je rovněž patrné, že není velký rozdíl mezi použitou technologií připojení. Rovněž musíme vzít v potaz i celkové vytížení databázového serveru, na kterém běží i ostatní hostující aplikace. Obecně lze ale konstatovat, že pro potřebný účel je rychlost odezvy v přijatelných a pro uživatele komfortních mezích a není třeba pro provoz databáze udržovat v provozu samostatný hardware.

8.2 Rychlost odezvy v komunikaci s řídicím serverem

Pro testování odezvy aplikace a řídicího serveru bylo zařízení prostřednictvím Wi-Fi připojeno do lokální sítě, kde se nacházel také řídicí server. Zaslání, zpracování a následná odpověď serveru se během 100 různých požadavků pohybovala mezi 80 a 120 ms. Nutno podotknout, že hodnota zahrnuje i odpověď serveru. Například pokyn pro zastavení motorického pohonu opon se od zaslání požadavku vykonal dříve, než dorazila samotná odpověď. Takto změřenou hodnotu považujeme za krajní, protože zbytek lokální sítě nebyl jinými zařízeními vytěžován.

Pro zajímavost byl řídicí server připojen přímo do internetu s otevřeným komunikačním portem. Požadavek byl poté zaslán skrze mobilní data přes celou veřejnou síť. Průměrná rychlost zaslání požadavku a následné odpovědi (pro 30 testů) byla 184 ms.

Závěr

Cílem práce bylo navrhnout a implementovat aplikaci, která by zastřešovala správu digitálního obsahu pro digitální kina dle specifikace DCI a zároveň řešila možnost bezdrátového ovládání a komunikaci se zařízeními zajišťující technické zabezpečení projekce (projektor, server, zvuková jednotka atd.).

Nejprve byla zevrubně provedena analýza současného stavu evidence a ovládání. Tato analýza byla použita k výběru vhodného řešení, kterým se stala aplikace pro mobilní platformu Android s napojením na on-line databázový server. Aplikace zároveň umožňuje bezdrátovou komunikaci s řídicím serverem pro ovládání jednotlivých zařízení.

Po datové analýze parametrů DCP balíků byla navržena struktura databáze a datový slovník. Pro provoz databáze byl vybrán komerční webový hosting s databázovým systémem MySQL a podporou skriptovacího jazyka PHP. Komunikace s databázovým serverem probíhá prostřednictvím zasílání HTTPS požadavků na aplikační server postavený na frameworku Nette a jazyce PHP.

Samotná aplikace pro tablet je napsána v objektově orientovaném jazyce Java. Aplikace je rozdělena na dvě části. První z nich se prostřednictvím již zmíněných HTTPS požadavků dotazuje na aplikační server a prostřednictvím něj vybírá data z databáze. Aplikace rovněž dokáže data třídit, zobrazovat relace mezi nimi a v neposlední řadě je do databáze vkládat upravovat a také mazat. Druhá část pak dle interakce uživatele a GUI zasílá HTTP požadavky na řídicí server, který je dále předává na koncová zařízení, zpracovává jejich odpověď a vrací ji zpět do aplikace.

Povedlo se navrhnout a implementovat funkční podobu aplikace a databáze se základními a pokročilejšími funkcionalitami. Bylo provedeno odzkoušení aplikace a komunikace s databází.

V nejbližší době se počítá s nasazením systému v testovacím provozu Kina Orion v Hradci nad Moravicí. Po dostatečném otestování v reálném provozu a odstranění případných nedostatků, se může systém rozšířit i do dalších jednosálových kin, která již o něj projevila zájem.

Díky použití on-line databázového serveru a přístupu skrze aplikační nastavbu je možné navržený systém evidence rozšířit o klientské aplikace i na jiné platformy (Windows, iOS, web atd.). Taková nastavba a zdokonalení současného systému může řešit navazující diplomová práce.

Literatura

- [1] DIGITALIZACE ČESKÝCH KIN: SNAD JEŠTĚ NENÍ POZDĚ. CINEPUR: časopis pro moderní cinefily [online]. červen 2008 [cit. 2019-02-05]. Dostupné z: cinepur.cz/article.php?article=1474
- [2] ŘEZNÍČKOVÁ, Aneta. Jednosálová kina se dokázala prosadit. Musela však investovat do techniky. IDnes.cz: Ekonomika [online]. MAFRA, 2018, 3.4.2018 01:00, 2018 [cit. 2018-12-27]. Dostupné z: https://www.idnes.cz/ekonomika/domaci/mala-kina-vs-multiplexy-prezila-a-zacinaji-rust-investice.A180402_204846_ekonomika_lre
- [3] PASEKOVÁ, Lucie. FENOMÉN MULTIKINO. Brno, 2011. Magisterská diplomová práce. MASARYKOVA UNIVERZITA. Vedoucí práce Mgr. Viktor Pantůček.
- [4] VACULA, Richard. Historie kina. Kino Orion - Hradec nad Moravicí: archivní i digitální projekce v rekonstruovaném jednosálovém kině z roku 1927 [online]. Hradec nad Moravicí: Postřeh, 2015, 4.12.2015 [cit. 2018-12-30]. Dostupné z: <https://www.kinoorion.cz/historie-kina/>
- [5] BŮBAL, Jan, Aleš DANIELIS, Richard FIALA, et al. ABC pracovníků kin: Pro interní potřebu pracovníků filmové distribuce odbor pro kádrovou a personální práci Ústřední půjčovny filmů, 111 21 Praha1, Národní 28. 4. přepracované vydání. Praha: Ústřední půjčovna filmů, 1987. ISBN 791.6/153/ABC.
- [6] Questions to ask your new TMS provider: Questions you should ask a prospective software provider. Arts Alliance Media [online]. 17. 7. 2017 [cit. 2019-01-24]. Dostupné z: <https://www.artsalliancemediacom.com/blog/questions-to-ask-your-new-tms-provider>
- [7] BECK, Ondřej. DCI projektory. Digitální kino: nezávislý web pro podporu zavádění digitálních technologií do českých kin [online]. Praha: Pro-DIGI o.s [cit. 2019-01-24]. Dostupné z: <http://www.digitalnikino.cz/node/343>
- [8] DP2K-10S: Small DLP Barco Alchemy Cinema projector (9,000 lumens) [online]. In: . BARCO, 2018, 20.12.2018, s. 1-4 [cit. 2019-01-24]. Dostupné z: <https://www.barco.com/services/website/en/ProductSpecSheet/ProductSpecSheetFile?productId=f38f703-44b0-842f-a746db19cfe6>
- [9] Christie CP2215 DLP digital cinema projector: Part Number: 127-010113-XX. In: CHRISTIE [online]. USA: Christie Digital Systems, 2019 [cit. 2019-02-24]. Dostupné z: <https://www.christiedigital.com/emea/cinema/cinema-products/digital-cinema-projectors/Christie-CP2215-DLP-Digital-Cinema-Projector>

- [10] SX-3000 Standalone Integrated Media Block™. GDC Technology Limited: Powering Your Digital Cinema Experience [online]. GDC Technology Limited, 2019, 18 February 2019 [cit. 2019-02-24]. Dostupné z: http://www.gdc-tech.com/english/products_sx3000_enterprise_storage.php
- [11] Model CP650 Digital Cinema Sound Processor: Installation Manual. 2. Dolby Laboratories, Inc, 2001. Dostupné také z: <http://www.iceco.com/dolby/CP650%20Installation%20Manual.pdf>
- [12] AP20 Audio Processor: Installation and Operating Guide. Datasat Digital Entertainment 9631 Topanga Canyon Place Chatsworth, CA 91, 2013. Dostupné také z: <http://datasatsupport.com/wp-content/uploads/2016/03/AP20-Users-Guide.pdf>
- [13] Communicator Touch Panel: Multi-user fingertip control. BARCO [online]. Beneluxpark 21, 8500 Kortrijk Belgium: BARCO, 2019 [cit. 2019-02-24]. Dostupné z: <https://www.barco.com/en/product/communicator-touch-panel>
- [14] MEO 5 XB1. WWW.MEOPTAHISTORY.COM [online]. Praha: Meopta-optika, 2011 [cit. 2019-02-05]. Dostupné z: www.meoptahistory.com/index.php?id=118
- [15] JNIOR Series 4: Getting Started Manual. In: INTEG [online]. 2919 E. Hardies Road, Gibsonia, PA 15044: INTEG Process Group, May 25, 2018 [cit. 2019-03-17]. Dostupné z: http://www.integpg.com/downloads/documentation/Getting_Started_Manual_Series_4.pdf
- [16] Products: Controllers. CUE [online]. Praha: CUE [cit. 2019-03-17]. Dostupné z: <https://www.cuesystem.com/controllers.aspx>
- [17] ECNA-10: D-Cinema Automation System. In: [Http://public.eprad.com/](http://public.eprad.com/) [online]. [cit. 2019-03-17]. Dostupné z: <http://public.eprad.com/literature/Automation/eCNA-10%20Automation%20Brochure.pdf>
- [18] Melody TMS by CinemaNext. In: CinemaNext [online]. YMAGIS, 2019 [cit. 2019-03-17]. Dostupné z: <https://cinemanext.com/melody-tms-cinemanext#>
- [19] TMS-2000 Theatre Management System: Technical Specifications. GDC Technology Limited: Powering Your Digital Cinema Experience [online]. 39 Healthy Street East, North Point, Hong Kong: GDC Technology Limited, 2019, 23 May 2018 [cit. 2019-03-17]. Dostupné z: http://www.gdc-tech.com/english/products_tms2000spec.php
- [20] Hypertext Transfer Protocol. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-03-17]. Dostupné z: https://cs.wikipedia.org/wiki/Hypertext_Transfer_Protocol
- [21] Úvod do JSON. Introducing JSON [online]. [cit. 2019-03-17]. Dostupné z: <https://www.json.org/json-cz.html>

- [22] Digital Cinema Naming Convention. In: Inter-Society Digital Cinema Forum (ISDCF) [online]. USA, 2019, 2019-02-20 [cit. 2019-04-09]. Dostupné z: <https://isdcf.com/dnc/>
- [23] DCI. Digital Cinema System Specification. USA: Digital Cinema Initiatives.
- [24] Promítač: Základní informace o způsobilosti promítače. UNIE FILMOVÝCH DISTRIBUTORŮ [online]. Praha, 28.8.2010 [cit. 2019-02-05]. Dostupné z: www.ufd.cz/clanky/promitac
- [25] ČSN EN 1838. Nouzové osvětlení (I): Světlo a osvětlení - Nouzové osvětlení. 2000.
- [26] Android 4.4 APIs: API Level: 19. Developers Android: build anything on android [online]. 23.01.2019 [cit. 2019-02-14]. Dostupné z: <https://developer.android.com/about/versions/android-4.4>
- [27] KOŠÁREK, Lukáš. Výkonnostní srovnání relačních databází. Brno, 2010. Bakalářská práce. Masarykova univerzita. Vedoucí práce RNDr. Vlastislav Dohnal, Ph.D.
- [28] Atomicity Consistency Isolation Durability (ACID): Definition - What does Atomicity Consistency Isolation Durability (ACID) mean?. Techopedia [online]. 2019 [cit. 2019-02-24]. Dostupné z: <https://www.techopedia.com/definition/23949/atomicity-consistency-isolation-durability-acid>
- [29] KANISOVÁ, Hana a Miroslav MÜLLER. UML srozumitelně. 2. aktualiz. vyd. Brno: Computer Press, 2006. ISBN 978-80-251-1083-6.
- [30] WELLING, Luke a Laura THOMSON. MySQL: průvodce základy databázového systému. Brno: CP Books, 2005. ISBN 80-251-0671-3.
- [31] OPPEL, Andrew J. Databáze bez předchozích znalostí: [průvodce pro samouky]. Brno: Computer Press, 2006. ISBN 80-251-1199-7.
- [32] POKORNÝ, Jaroslav a Ivan HALAŠKA. Databázové systémy. Vyd. 2. přeprac. Praha: Vydavatelství ČVUT, 2003. ISBN 80-010-2789-9.
- [33] VÁVRŮ, Jiří a Miroslav UJBÁNYAI. Programujeme pro Android. 2., rozš. vyd. Praha: Grada, 2013. Průvodce (Grada). ISBN 978-80-247-4863-4.
- [34] HERODEK, Martin. Android: jednoduše. 2. aktualiz. vyd. Brno: Computer Press, 2014. Naučte se za víkend (Computer Press). ISBN 978-80-251-4298-1.
- [35] GERBER, Adam a Clifton CRAIG. Learn Android Studio: build Android apps quickly and effectively. Berkeley, CA: Apress, [2015]. ISBN 978-1430266013.
- [36] POKORNÝ, Jaroslav a Ivan HALAŠKA. Databázové systémy: vybrané kapitoly a cvičení. 2. přeprac. vyd. Praha: Karolinum, 1998. ISBN 80-718-4687-2.

- [37] BLOOM, Jeffrey. Digital Cinema Content Security and the DCI. 2006 40th Annual Conference on Information Sciences and Systems. IEEE, 2006, 2006, , 1176-1181. DOI: 10.1109/CISS.2006.286643. ISBN 1-4244-0350-2. Dostupné také z: <http://ieeexplore.ieee.org/document/4067984/>
- [38] ECKHARDTOVÁ, Dita. Bez drátů ve zkratce: Bluetooth versus WiFi: (Seznamte se s 'Wireless'). In: Technet.cz [online]. Praha: Mafra, 2019, 19. ledna 2004 0:01 [cit. 2019-04-09]. Dostupné z: https://www.idnes.cz/technet/pc-mac/bez-dratu-ve-zkratce-bluetooth-versus-wifi-seznamte-se-s-wireless.A040119_5250250_tech-a-trendy-nb

A Blokové schéma instalace systému

Blokové schéma instalace systému je vytištěno na formát A4 a vloženo do kapsy na přílohy. Rovněž se nachází na přiloženém CD a v elektronické příloze pod názvem *A Blokové schéma instalace systému.pdf*.

B Aktivita diagram mobilní aplikace

Aktivita diagram mobilní aplikace je vytištěn na formát A3 a vložen do kapsy na přílohy. Rovněž se nachází na přiloženém CD a v elektronické příloze pod názvem *B Aktivita diagram mobilní aplikace.pdf*.

C Video-ukázky

C.1 Spuštění a přihlášení uživatele do aplikace

Video-ukázka **Spuštění a přihlášení uživatele do aplikace** se nachází na přiloženém CD a v elektronické příloze v adresáři *C Video-ukázky* pod názvem *C.1 Spuštění a přihlášení uživatele.mp4*

C.2 Prohlížení záznamů DCP z databáze

Video-ukázka **Prohlížení záznamů DCP z databáze** se nachází na přiloženém CD a v elektronické příloze v adresáři *C Video-ukázky* pod názvem *C.2 Prohlížení záznamů DCP z databáze.mp4*

C.3 Prohlížení záznamů KDM z databáze

Video-ukázka **Prohlížení záznamů KDM z databáze** se nachází na přiloženém CD a v elektronické příloze v adresáři *C Video-ukázky* pod názvem *C.3 Prohlížení záznamů KDM z databáze.mp4*

C.4 Vložení nového DCP záznamu do databáze

Video-ukázka **Vložení nového DCP záznamu do databáze** se nachází na přiloženém CD a v elektronické příloze v adresáři *C Video-ukázky* pod názvem *C.4 Vložení nového DCP záznamu do databáze.mp4*

C.5 Vložení nového KDM záznamu do databáze

Video-ukázka **Vložení nového KDM záznamu do databáze** se nachází na přiloženém CD a v elektronické příloze v adresáři *C Video-ukázky* pod názvem *C.5 Vložení nového KDM záznamu do databáze.mp4*

C.6 Úprava stávajícího DCP záznamu v databázi

Video-ukázka **Úprava stávajícího DCP záznamu v databázi** se nachází na přiloženém CD a v elektronické příloze v adresáři *C Video-ukázky* pod názvem *C.6 Úprava stávajícího DCP záznamu v databázi.mp4*

C.7 Úprava stávajícího KDM záznamu v databázi

Video-ukázka **Úprava stávajícího KDM záznamu v databázi** se nachází na přiloženém CD a v elektronické příloze v adresáři *C Video-ukázky* pod názvem *C.7 Úprava stávajícího KDM záznamu v databázi.mp4*

C.8 Odstranění stávajícího DCP záznamu z databáze

Video-ukázka **Odstranění stávajícího DCP záznamu z databáze** se nachází na přiloženém CD a v elektronické příloze v adresáři *C Video-ukázky* pod názvem *C.8 Odstranění stávajícího DCP záznamu z databáze.mp4*

C.9 Odstranění stávajícího KDM záznamu z databáze

Video-ukázka **Odstranění stávajícího KDM záznamu z databáze** se nachází na přiloženém CD a v elektronické příloze v adresáři *C Video-ukázky* pod názvem *C.9 Odstranění stávajícího KDM záznamu z databáze.mp4*

C.10 Ovládání zařízení

Video-ukázka **Ovládání zařízení** se nachází na přiloženém CD a v elektronické příloze v adresáři *C Video-ukázky* pod názvem *C.10 Ovládání zařízení.mp4*

D Zdrojové kódy a instalační soubory

D.1 Skript pro vytvoření databáze

Skript pro vytvoření databáze se nachází na přiloženém CD a v elektronické příloze v adresáři *D Zdrojové kódy a instalační soubory* pod názvem *D.1 Skript pro vytvoření databáze.sql*

D.2 Zdrojový kód aplikačního serveru

Zdrojový kód aplikačního serveru **baseCinema** se nachází na přiloženém CD a v elektronické příloze v adresáři *D Zdrojové kódy a instalační soubory**D.2 Zdrojový kód aplikačního serveru* jako adresář s názvem *baseCinema*

D.3 Zdrojový kód mobilní aplikace

Zdrojový kód mobilní aplikace **conCinema** se nachází na přiloženém CD a v elektronické příloze v adresáři *D Zdrojové kódy a instalační soubory**D.3 Zdrojový kód mobilní aplikace* jako adresář s názvem *conCinema*

D.4 Instalační soubor mobilní aplikace

Instalační soubor mobilní aplikace **conCinema** se nachází na přiloženém CD a v elektronické příloze v adresáři *D Zdrojové kódy a instalační soubory**D.4 Instalační soubor mobilní aplikace* pod názvem *conCinema.apk*